How dynamical regime and neuronal network structure influence synchronous events

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Brittany Paige Baker

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Duane Nykamp

June, 2019

Acknowledgements

The people with which I have surrounded myself throughout this journey have played a vital role in my ability to complete this dissertation.

First and foremost, I am extremely grateful for my adviser, Duane Nykamp. He provided endless encouragement, guidance, and patience over the past few years. His support has been the most important factor in my success and without him this would not have been possible. I would also like to thank Sam Fuller for her work on the subject, and her suggestion to approach Duane for advising when I needed a change in the direction of my graduate career.

I would also like to thank my friends: Stef Hornung for initiating a sense of home when I was new to Minnesota, and being the most consistent and loyal friend that reminds me that there's more to life than mathematics; Amy Doucette Gaskin for being a long-term loyal friend that brings the comfort and reminder of shenanigans from home; Samantha Schumacher for being my unofficial mentor and friend throughout graduate school; Jeffery Moulton, Shannon Negaard-Paper, and Adam Maidman for their stimulating conversations and support; Monika Hening and Laurie deCastro for staying in touch over the years and providing mathematical support from afar; Fedor Korsakov and David Cook for challenging my mind with outside puzzles, games, and conversations; Molly Nelson, Kelly Lewis, and other Jazzercise ladies that have helped me decompress from long days and weeks of mathematics; Amber Yuan for fantastic meals and Costco trips; Greg Michel, Harini Chandramouli, and Mike Loper for hours of accountability work in coffee shops; Jacob Miller for his unique ability to understand where I come from, where I currently am, and where I may be going, and providing continuous acceptance and encouragement; Salina Renninger for always listening and helping me work through my many feelings and emotions of the past three years; George Ashline, Zsu

Kadas, Jo Ellis-Monaghan, Lloyd Simons, and Greta Pangborn, professors of mathematics and computer science at Saint Michael's College, my undergraduate institution, for staying in contact and providing continued support, words of encouragement, and advise prior to and throughout my graduate career; Ryan Coopergard, Elizabeth Kelley, Sarah Milstein, and Tara Palnitkar for being the most supportive and understanding office mates to surround myself with as I complete this journey.

Finally, the love and support that I have received from my family cannot be exaggerated. They believed in me even when I didn't believe in myself. Their encouragement helped me carry on through the good times and the bad. Thus, I thank them for their constant confidence in my current and future success.

Abstract

Synchronization of spiking neuronal activity plays a role in many important processes in the human body. Zhao et al. [1] explored the relationship between synchrony and network structure by developing the SONET model where one can modulate the microstructure of the network by adjusting frequencies of pairs of directed connections between nodes, which correspond to the second order statistics of the network. We extended the SONET framework to allow for the prescription of probabilities of neuronal connections based on location to modulate spatial macrostructure. We used this spatial SONET model to explore how both network microstructure (SONET motif frequencies) and macrostructure influence the emergence of synchrony. To enable a consistent analysis of synchrony across a wide range of networks, we developed a novel measure of synchrony based on the rate of synchronous events. We discovered that the microstructure played the dominant role in shaping synchrony. Moreover, we found that the influence of the microstructure can depend on the dynamics of the inputs to the network.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The nervous system is an extensive and complex network of neurons. Perin et al. [2] and Song et al. [3] both observed that the structure of networks of neurons is more clustered than random. We utilize SONET motif frequencies introduced by Zhao et al. [1] to explore how they may influence the synchrony of networks with various macrostructures. We also considered how dynamical regime modulates the influence of SONET motif frequencies on synchrony.

- Chapter 2 provides neuroscience and neuronal network theory background and discusses previous work to provide motivation.
- Chapter 3 explains the network model and simulation methods that we used. We describe the SONET motif model and various macrostructures of neuronal networks and we give an overview of our network generation algorithm. Our simulation model is discussed and we explain how we developed our measure for synchrony.
- Chapter 4 analyzes and discusses our results. We examine networks with varying microstructure and macrostructure. A description of distinct dynamical regimes is introduced. We evaluate and discuss the synchrony of numerous spiking neuron simulations.
- Chapter 5 introduces a stochastic simulation model that we imposed to seek more understanding of the factors that may effect the synchrony of a network.

Chapter 2

Background and motivation

We begin by giving an overview of the neuroscience background and random network theory needed to discuss the effects of neuronal network structure on network dynamical patterns such as synchrony. We also discuss the importance of synchrony and consider previous work and observations that motivate our work.

2.1 Background

2.1.1 Neurons and the nervous system - anatomy

Most animals have a nervous system to detect environmental conditions that affect the body. The nervous system is composed of cells called neurons that use electrochemical signals and neurotransmitter chemicals to transmit signals within the brain as well as between the brain and the rest of the body. The nervous system is extremely complex, and is composed of billions of neurons and trillions of connections called synapses.

A typical neuron consists of a cell body (soma), dendrites, and an axon. The axon and dendrites are tendrils that extend from the compact soma. In humans, the axon could be up to 1 meter in length, though usually relatively small and constant in diameter. At the end of an axon are axon terminals, which transmit electrochemical signals to another cell. Dendrites, which are usually much shorter than the axon, receive electrochemical stimulation from other cells.

The cell membrane of a neuron is a phospholipid bilayer that is an electrical insulator.

Embedded in the membrane are proteins. These proteins form ion channels and pumps that allow for ions to be transported from one side of the membrane to the other. Some ion channels are voltage gated, while others are chemically gated. The movement of these ions alter the voltage across the membrane.

2.1.2 Neuronal activity

Neurons communicate to each other via synapses, the area where an axon terminal of one neuron connects to another neuron, typically at a dendrite. Neurons fire when the electric voltage reaches or exceeds a threshold and travels as an electric pulse (action potential) along the axon of the neuron. This causes the axon terminal to release neurotransmitters into the synapse. These neurotransmitters then latch onto receptors in the dendrite of a nearby neuron, causing more channels to open and the voltage of that neuron to change.

The neuron that releases neurotransmitters is called a presynaptic neuron, and the neuron that receives neurotransmitters is called a postsynaptic neuron. If external input causes the membrane potential of a neuron to become sufficiently high, then it will activate an action potential that will travel as a pulse down the axon of the neuron, and the firing process will repeat.

2.1.3 Neural synchrony

Synchrony is the term we use to describe the phenomena when many of the neurons in a network fire at the same time or consecutively. Some consider synchrony to be the neural indicator of consciousness, and that it plays a vital role in memory formation. Synchrony and neuronal network structure have been found to play an important role in memory formation [4], and may also be related to many neurological diseases such as schizophrenia [5], Parkinson's disease [6], Alzheimer's disease [7], and epilepsy [8, 9].

Synchronization is believed to be caused by many factors such as network structure, individual neuron dynamics, the types of synaptic response, and even the network environment and dynamics of the inputs to the network. We have explored how local and global network structure as well as dynamical regime (input dynamics) may impact synchronization. Our findings are described in Chapter 4.

2.1.4 Network theory

A random network is a probability distribution over a graph, often simply described by the probability distribution that generates it. The most basic type of random graph was introduced by Paul Erdős and Alfréd Rényi [10]. In an Erdős-Rényi (ER) random network, each edge between any pair of nodes is independent of all others and equally likely to occur.

The ER model does not take any factors, such as node location, into consideration when assigning edges. Thus, an ER random network can be completely characterized by the number of nodes (N), and the probability of connection between any two nodes (p), which describes the first order statistic of the random network. This independent random network model is a natural choice when beginning to study neuronal networks, as it has minimal structure beyond what is required by the connection probabilities.

However, a study by Perin et al. [2] found that cells in neocortical microcircuits are not arranged randomly, but as small world networks where the connection probability between two neurons is proportional to their number of common neighbors. Similarly, Song et al. [3] investigated the frequencies of sixteen different motifs on triplets of neurons. They discovered that bidirectional connections were much more common than expected, and that the strengths of synaptic connections sharing pre- or postsynaptic neurons were correlated, concluding, as Perin et al. did, that strong connections are more clustered than weak.

Therefore, we base our work on the Second Order Network (SONET) model, introduced by Zhao et al. [1], to add structure beyond the independent Erdős-Rényi model by modulating the frequencies of pairs of directed connections between nodes. This alters the second order statistics of the network, and adds low-dimensional structure to the network. The four second order connection motifs (reciprocal, convergent, divergent, and chain) are illustrated in Figure 3.1, and the SONET model is described in detail in section 3.1.1.

2.2 Previous work

2.2.1 SONETS

When Zhao et al. introduced the SONET model, they included a framework to relate network connectivity to synchrony in spatially homogeneous neuronal networks. They observed an influence of second order connectivity on synchrony through chain and convergent connection motifs, and discussed how synchrony tended to increase with the relative frequency of chain connection motifs and decrease with the relative frequency of convergence connection motifs. They also illustrated their observation that the synchrony of a network appeared unaffected by the relative frequency of divergent connection motifs. Later, Nykamp et al. [11] developed mean-field equations that explain these observations.

2.2.2 Feed-forward networks

Liu and Nykamp [12] developed a kinetic theory approach to capturing interneuronal correlation in layered feed-forward networks, where neurons are clustered in layers and the neurons in a layer only receive inputs from neurons in a previous layer. In this study, they observed that the average number of neurons from a population that simultaneously project onto a pair of neurons (divergence) leads to a buildup of correlations, which tends to cause synchrony.

Rosenbaum et al. [13, 14] also examined the development of synchrony in layered feed-forward networks. They observed that convergent connections, which they referred to as pooling, increase correlation and that the development of synchrony is primarily due to this pooling of correlated inputs. They also examined the effects of divergence, which they referred to as overlap, in down-stream correlation, but concluded that this did not have as much of an impact as pooling.

We expand upon these ideas by focusing on feed-forward networks without layers, but with a spacial structure and varying frequencies of SONET motifs. We conduct spiking neuron simulations using a leaky integrate-and-fire model and observe the effects of dynamical regime and SONET motif frequencies on synchrony.

Chapter 3

Network model and simulation methods

3.1 Network model

Zhao et al. [1] developed a second order network (SONET) model that allows for the generation of neuronal networks with specified frequencies of the second order connection motifs shown in Figure 3.1. Their model was restricted to spatially homogeneous networks, where the probability of connection was constant for all pairs of neurons. Hence, the SONET framework is unable to model networks with spatial clusters of neurons. Following the work of Fuller [15], we extend the homogeneous SONET model to include a spatial component with connection probability dependent on neuron location.

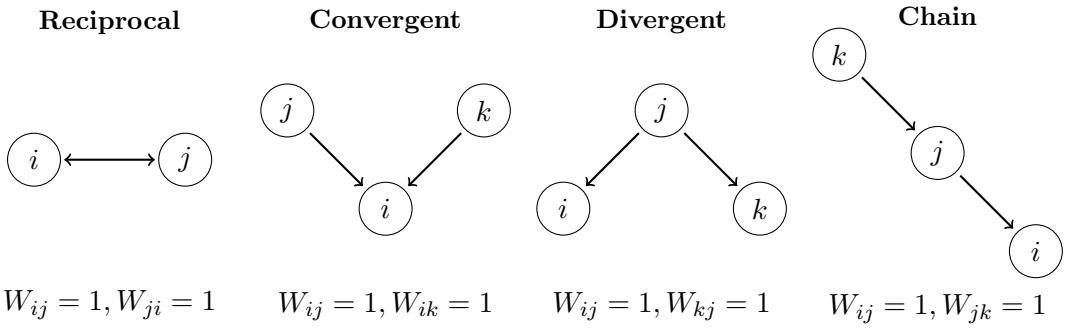


Figure 3.1: The four second order connection motifs.

3.1.1 Microstructure via SONET motif frequencies

We let W represent the connectivity matrix of the network, then $W_{ij} = 1$ if there is a connection from neuron j to neuron i, and 0 otherwise. Further, we do not allow self-coupling, so $W_{ii} = 0$ for all i. We develop W based on second and first order statistics. First, because we are taking the one-dimensional "location" of a neuron into account, we consider the first-order connection probability of the matrix $P(W_{ij} = 1) = E(W_{ij}) = p_{ij}$, which depends on the particular pair of neurons.

We also consider the four distinct patterns of pairs of connections sharing a common neuron. We call these reciprocal, convergent, divergent, and chain motifs (see Figure 3.1). These motifs are associated with the second-order statistics of the network. We allow the first-order statistics of the network to depend on neuron index (location), but for simplicity, we keep the second-order statistics independent of neuron index. Although we do not have data to determine the biological validity of this, we consider it reasonable to assume that the second-order statistics are independent of neuron location. Also, this assumption makes our model tractable.

To allow us to prescribe these second-order statistics then generate networks, we introduce the variables $\alpha_{\text{recip}}, \alpha_{\text{conv}}, \alpha_{\text{div}}$, and α_{chain}, which measure deviations from independence of the frequency of the four two-edge motifs.

$$
\begin{aligned}
P(W_{ij}=1, W_{ji}=1) &= p_{ij}p_{ji}(1+\alpha_{\text{recip}}) & & & \alpha_{\text{recip}} &= \frac{\text{cov}(W_{ij}, W_{ji})}{p_{ij}p_{ji}} \\
P(W_{ij}=1, W_{ik}=1) &= p_{ij}p_{ik}(1+\alpha_{\text{conv}}) & & & \alpha_{\text{conv}} &= \frac{\text{cov}(W_{ij}, W_{ik})}{p_{ij}p_{ik}} \\
P(W_{ij}=1, W_{kj}=1) &= p_{ij}p_{kj}(1+\alpha_{\text{div}}) & \text{equivalently,} & & \alpha_{\text{div}} &= \frac{\text{cov}(W_{ij}, W_{kj})}{p_{ij}p_{kj}} \\
P(W_{ij}=1, W_{jk}=1) &= p_{ij}p_{jk}(1+\alpha_{\text{chain}}) & & & \alpha_{\text{chain}} &= \frac{\text{cov}(W_{ij}, W_{jk})}{p_{ij}p_{jk}}.
\end{aligned}
$$

These α values are now parameters that we can alter to generate desired networks.

Second-order statistics in relation to degree distribution

To give additional intuition on the nature of these statistics, we can relate some of these second-order statistics to the in-degree and out-degree distributions of the network. The

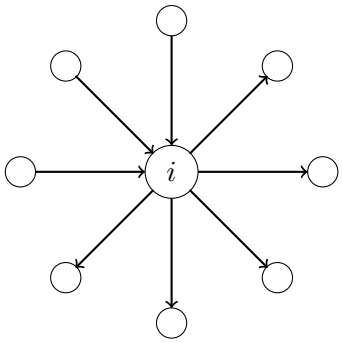


Figure 3.2: An illustration of a node with an in-degree of three and an out-degree of 5.

in-degree of a neuron is the number of neurons that connect to it, and the out-degree is the number of neurons that it connects to. In Figure 3.2, we see a simple network where neuron i has an in-degree of 3 and an out degree of 5.

We denote the in-degree of neuron i by $d_i^{\text{in}} = \sum_j W_{ij}$, and the out-degree by $d_i^{\text{out}} = \sum_j W_{ji}$. Obviously, the mean in-degree is equivalent to the mean out-degree, which we will simply call the mean degree and denote as $d = \frac{1}{N}\sum_{i,j} W_{ij}$, where N is the number of neurons in the network.

Then, from Zhao et al. [1], when N is large, we get the following relationships between second-order statistics and degree distribution:

$$\begin{aligned}
\alpha_{\text{conv}} &\approx \frac{\text{var}(d_i^{\text{in}}) - E(d)}{E(d^2)} \\
\alpha_{\text{div}} &\approx \frac{\text{var}(d_i^{\text{out}}) - E(d)}{E(d^2)} \\
\alpha_{\text{chain}} &\approx \frac{\text{cov}(d_i^{\text{in}}, d_i^{\text{out}})}{E(d^2)}.
\end{aligned}$$

This gives us intuition of the effect of the network connectivity statistics on the synchrony of the network.

3.1.2 Network generation algorithm

We can prescribe first order (p_{ij}) and second order ($\alpha_{\text{recip}}, \alpha_{\text{conv}}, \alpha_{\text{div}}$, and α_{chain}) statistics of the network, but those do not uniquely determine a network. We likely have an underdetermined system as there could be many probability distributions with those

statistics. As a way to choose among those possible probability distributions, we seek the one with the least structure, or the maximum entropy, which is difficult to find. Therefore, we will use a model of dichotomized Gaussian (sometimes called near maximum entropy) [16].

Alternatively, there may be no probability distribution with the prescribed first and second order statistics. Or, the prescribed model may not actually represent a real graph. For example, it is entirely possible (and relatively easy) to produce a symmetric matrix that is not positive definite, and therefore cannot be a covariance matrix of a network. Thus, our algorithm is designed to attempt to generate a network from a particular model and quickly determine if the prescribed parameters are impossible. These considerations were present in the original SONET model, but became much trickier to achieve when a spatial component was added. In particular, we are interested in cases where the connection probability strongly depends on space, which makes network generation challenging.

Our approach is an extension of the spatially homogenous SONET model by Zhao et al. [1]. This generalization of SONETs was begun by Fuller [15], who developed a framework to add modulation of connection probability (p_{ij}) in the presence of SONET microstructure. Although this initial model was limited in the amount of possible variation in p_{ij} and did not scale well with network size, it laid out a successful approach for adding spatial structure to the SONET model. We extend Fuller's model to regimes of strong spatial dependence and improved its scaling with network size so that we can easily generate networks with tens of thousands of nodes.

Thus, to create networks, we generate the connectivity matrix W. We use a dichotomized Gaussian approach to add minimal structure beyond the desired first and second order statistics (minimize higher order statistics). To do this we generate Gaussian random variables Z_{ij}, with specified covariance matrix Σ. Then we threshold each Z_{ij} to create the connectivity matrix W:

$$W_{ij} = \begin{cases} 1, & \text{if } Z_{ij} > \theta_{ij} \\ 0, & \text{otherwise,} \end{cases}$$

where the threshold value θ_{ij} is chosen such that $P(W_{ij} = 1) = p_{ij}$. Thus, $\theta_{ij} = \infty$ if

$p_{ij} = 0$. The covariance matrix Σ is such that we get the desired first and second order statistics of W.

For even moderate-sized networks, Σ is too large to fit in memory, as it is size $N^2 \times N^2$. So we exploit its spatial structure to tractably generate random variables, Z_{ij} with that covariance. The covariance matrix Σ has a particularly simple structure for homogeneous networks, but is destroyed once we allow p_{ij} to depend on location. However, with some assumptions on p_{ij}, and an number of approximations, we can generalize the SONET approach to a spatially varying p_{ij}.

We follow this process to generate W:

1. Determine the covariance matrix Σ of the Z_{ij} Gaussian random variables. Σ is such that $\mathrm{Var}(Z_{ij}) = 1$, and yields the appropriate second order statistics for W. Thus, Σ depends on p_{ij} and α values.

2. Compute an approximate square root S of Σ. Note that, even for relatively small networks, both Σ and S may be too large for memory, so we exploit their structure and store components in terms of a small number of parameters.

3. Generate a matrix X of standard normal random variables, then let $Z = SX$.

4. Finally, threshold each Z_{ij} to get the connectivity matrix W.

This algorithm operates under the assumptions that there exists a square root S of the covariance matrix Σ, that is structured similarly to Σ. We make this assumption because S, like Σ, is too large to fit in memory. We assume that they have similar structure so that we can use the same, much smaller number of parameters, to describe both. Further, it is assumed that no neuron connects to itself, so $P(W_{ii} = 1) = p_{ii} = 0$, for all i, and that $\sum_j p_{ij} = \sum_j p_{ji}$ is equivalent for all i. This last assumption is required to allow us to exploit the structure of S and Σ (see section A.4 for thorough justification).

Complete details of our network generation algorithm for prescribed first and second order statistics (p_{ij} and α values) can be found in Appendix A.

Estimating connectivity statistics

For spatially homogeneous SONETs, we can estimate these first and second order statistics directly from a given connectivity matrix W. We let $\hat{p}$ denote the average connectivity of the matrix. Suppose $N_{\text{conn}} = \|W\|_1 = \sum_{i,j} W_{ij}$ is the number of connections in W, then

$$\hat{p} = \frac{N_{\text{conn}}}{N(N-1)}.$$

Further, we can compute the second order statistics of the matrix with the following relations:

$$\begin{aligned}
\hat{p}^2(1+\hat{\alpha}_{\text{recip}}) &= \frac{N_{\text{recip}}}{N(N-1)/2} \\
\hat{p}^2(1+\hat{\alpha}_{\text{conv}}) &= \frac{N_{\text{conv}}}{N(N-1)(N-2)/2} \\
\hat{p}^2(1+\hat{\alpha}_{\text{div}}) &= \frac{N_{\text{div}}}{N(N-1)(N-2)/2} \\
\hat{p}^2(1+\hat{\alpha}_{\text{chain}}) &= \frac{N_{\text{chain}}}{N(N-1)(N-2)}.
\end{aligned}$$

Here we let $N_{\text{recip}}, N_{\text{conv}}, N_{\text{div}}$, and N_{chain} represent the number of corresponding second order motifs in the network, each of which we can compute directly:

$$\begin{aligned}
N_{\text{recip}} &= \text{Tr}(W^2)/2 \\
N_{\text{conv}} &= \left(\|W^T W\|_1 - \|W\|_1\right)/2 \\
N_{\text{div}} &= \left(\|W W^T\|_1 - \|W\|_1\right)/2 \\
N_{\text{chain}} &= \|W^2\|_1 - \text{Tr}(W^2).
\end{aligned}$$

Hence,

$$\hat{\alpha}_{\text{recip}} = \frac{\text{Tr}(W^2)}{N(N-1)\hat{p}^2} - 1$$
$$\hat{\alpha}_{\text{conv}} = \frac{\|W^T W\|_1 - \|W\|_1}{N(N-1)(N-2)\hat{p}^2} - 1$$
$$\hat{\alpha}_{\text{div}} = \frac{\|W W^T\|_1 - \|W\|_1}{N(N-1)(N-2)\hat{p}^2} - 1$$
$$\hat{\alpha}_{\text{chain}} = \frac{\|W^2\|_1 - \text{Tr}(W^2)}{N(N-1)(N-2)\hat{p}^2} - 1.$$

Since our networks are computed probabilistically, these $\hat{\alpha}$ values likely differ from the prescribed α values that were used to generate the the network. Hence, we will use the $\hat{\alpha}$ values when analyzing the synchrony of the network.

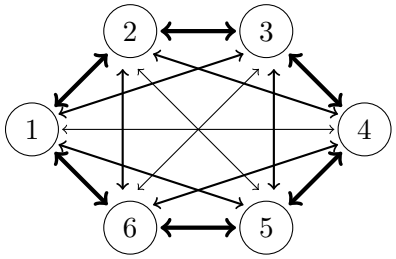
It is important to note that these approximations are accurate only for spatially homogeneous networks and not valid for networks with varying connection probabilities (p_{ij}). We use them anyway, as this is how we would estimate the α values if p_{ij} was unknown, though we acknowledge that they are not estimates of the actual α values. To estimate the actual α values of spatial networks, we would first have to estimate p_{ij}, which would require multiple samples of the network or assumptions on the form of p_{ij}. Although we could estimate p_{ij}, that cannot be done with a single sample (network). Hence, as our only option, we still use these $\hat{\alpha}$ values with spatial networks, as they appear to capture key deviations and provide useful information.

3.1.3 Macrostructure based on one-dimensional geometry

We consider a spatial component where neurons that are closer together are more likely to be connected, as this reflects a general structure that has been observed within the brain. We use a Gaussian kernel, so we let $p_{ij} = p_{\max} \exp\left(-d_{ij}^2/2\sigma^2\right)$, where d_{ij} is the distance between neuron i and neuron j, and $p_{\max}$ is a scalar so the average number of connections per neuron is a prescribed value, p_{avg}. We consider various values for σ to determine the spatial scale of connections (i.e., how close neurons need to be to each other to have a non-zero probability of being connected).

Using this spatial component, we considered two different macrostructures for our networks: a recurrent model where we allow neurons to be oriented in a ring, with

(a) Recurrent model



(b) Feed-forward model

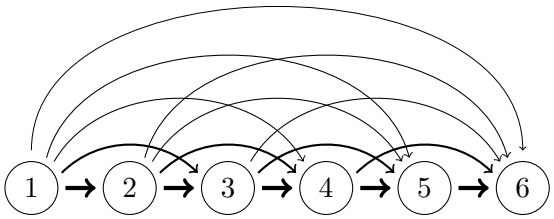


Figure 3.3: Simple examples of networks with two different macrostructures based on one-dimensional geometry. On the left we have a recurrent model with connections in both directions. On the right, a feed-forward network with connections moving forward through the network. Neurons that are closer together are more likely to be connected. Hence, line thickness indicates connection probability.

connections allowed in either direction around the ring (see Figure 3.3a), and a feed-forward model where we assume neurons are oriented on a line segment, with only forward connections allowed (see Figure 3.3b).

The assumption that $\sum_j p_{ij} = \sum_j p_{ji}$ is equivalent for all i (which allows us to exploit the structure of S and Σ) is still imposed as an initial step in the feed-forward case. Each feed-forward network is generated in a ring, then the network is truncated so that $W_{ij} = 0$ if $j \geq i$.

3.2 Simulation model

We use the Brian spiking neural network simulator [17] with Python. We run simulations on networks of excitatory neurons with instantaneous synapses using a leaky integrate-and-fire model:

$$\frac{dv_i}{dt} = -\frac{1}{\tau}[v_i(t) - v_{\text{rest}}] + \sum_j \sum_k J\, W_{ij}\, \delta(t - T_{j,k}) + \sum_k I^{\text{ext}}\, \delta(t - T^{\text{ext}}_{i,k}),$$

where

- $v_i(t)$ is the membrane potential of neuron i

- $J = 0.18$mV is the connectivity strength (the voltage that is passed 'downstream' when a neuron fires)
- $T_{i,k}$ is the time of kth spike of neuron i
- I^{ext} is the magnitude of the external input
- $T^{\text{ext}}_{i,k}$ is the time of the kth external input to neuron i, a Poisson process with rate λ
- $v_{\text{rest}} = -60$mV and $\tau = 10$ms are constants
- W_{ij} is the indicator of connectivity of neuron j to neuron i
- δ is the Dirac delta function.

In this model, a spike of neuron i occurs when $v_i(t) = v_{\text{thresh}} = -55$mV, after which $v_i(t)$ is set to $v_{\text{reset}} = -65$mV for a refractory period of 1ms.

We vary the magnitude, I^{ext}, and rate, λ, of external input to the system to establish different dynamical regimes for simulation. This is explained in detail in section 4.2.

3.3 Synchronous events

We seek to explore how synchrony depends on network microstructure (SONET motifs), network macrostructre (spatial dependence of connection probability), and dynamical regime (described in section 4.2). Typical synchrony measures include the Kuramoto order parameter [18] and measures based on autocorrelation or spectrum of population activity. Such measures include the assumption of periodicity: either the node dynamics are represented by oscillators or we must exploit bifurcations (such as Hopf bifurcation) in the dynamics of the population activity that leads to periodic behavior [11, 19, 20].

However, our goal is to explore macostructures (specifically feed-forward networks) for which these previous population models and resulting bifurcation analyses do not apply. Moreover, we are interested in dynamical regimes (in particular, the irregular regime described in section 4.2) where networks do not exhibit periodic behavior. Hence, the standard synchrony measures will not apply or give meaningful results. To obtain a

quantitative measure of synchrony that pertains to all of our simulations, we developed a new synchrony measure.

Our synchrony measure was motivated by observations of network behavior. We use Brian neuron spiking simulator [17] in Python to perform a simulation, then observe synchronous firings that travel in waves through the network. We call these events. Events can be visually recognized by examining the raster plot of a simulation. (A raster plot has vertical axis neuron index and horizontal axis time, with a dot at (i, t) if neuron i fired at time t.) In our feed-forward model synchronous events travel forward through the network, then stop when they reach the end of the network (see Figure 3.4). In recurrent networks, the events travel in both directions through the network, and wrap around, stopping only when they collide with themselves (see Figure 3.5). Our measure of synchrony will be the rate at which these events occur, the "event rate".

To determine when synchronous events occur, we look at the average activity level localized in time and space and look for instances where this is unusually high. We have observed that unusually high activity level usually propagates through the network (in a wave-like fashion) and consider this phenomenon a synchronous event. We also observed that this propagation of activity is not always continuous in time, so we allow for slight time gaps, but we do not allow for gaps in space.

The general procedure for identifying synchronous events is:

1. Calculate the average activity for groupings of neurons.
2. Identify episodes where neuron activity is unusually high.
3. Concatenate adjacent episodes into events.

3.3.1 Average activity matrix

The first step of our event detection algorithm is to measure the average activity of neighboring neurons at each moment in time. We begin by considering neurons to be in bins of $\Delta N = 100$ neurons. We also bin the simulation time by $\Delta t = 0.1$ms (this is done automatically with the Brian neuron spiking simulator). For each neuron bin and time bin we calculate the number of neuron firings averaged over time and space, and

we let A represent the matrix of this average activity. That is,

$$A_{ij} = \frac{\eta_{ij}}{\Delta N \Delta t},$$

where η_{ij} is the number of neuron firings that occur in neuron bin i and time bin j. We then look for entries in A that are larger than average. Since an event is an occurrence of unusually high activity, our next step is to determine an appropriate threshold for each neuron bin that can be used to detect when the activity is unusually high.

3.3.2 Threshold vector

The goal is to have a threshold value for each neuron bin that is reached only rarely. Thus, we define a threshold that depends on the proportion of time bins with neural activity.

We consider Bernoulli random variables X_{ij} that indicate whether or not we have at least one neuron from bin i that fires in time bin j. Thus, we let $P(X_{ij} = 1) = \beta_i$, where β_i be the fraction of time bins with at least one neuron from bin i that fired. We choose simulation parameters so that in most time and neuron bins, we do not have a neuron fire, therefore that for each neuron bin i the median of $\{X_{ij}\}_{j \geq 0}$ is 0. We then let the threshold value for activity in neuron bin i be γ_i, which is set to 15 standard deviations above the median, scaled over time and space:

$$\gamma_i = \frac{15\sqrt{\beta_i(1-\beta_i)}}{\Delta N \Delta t}.$$

3.3.3 Identifying episodes

Once we have determined a threshold for a given neuron bin, the next step is to identify episodes where that neuron bin's activity is unusually high. Roughly speaking, an episode is an interval of time $(t_{\text{start}}, t_{\text{end}})$ during which A_{ij} is large for $t_{\text{start}} \leq j \leq t_{\text{end}}$. To account for variability in the data, we adopt the following, slightly relaxed, definition of an episode:

1. The episode requires either very high activity ($A_{ij} \geq 3\gamma_i$) in a single time bin, or sustained high activity ($A_{ij} \geq \gamma_i$) for at least three time bins.

2. The episode may contain gaps of lower activity ($A_{ij} < \gamma_i$) as long as each gap is two time bins or shorter.

3. The activity is low ($A_{ij} < \gamma_i$) for the three time bins preceding and following the episode.

3.3.4 Concatenate episodes into events

To identify synchronous events, we look for sequences of episodes that travel across neighboring neuron bins. We separately identify synchronous events that travel up the network (neuron bin increasing with time), and events that travel down the network (neuron bin decreasing with time). To qualify as an event, a sequence of episodes must cover at least three adjacent neuron bins. Further, an event cannot skip a neuron bin, but we do allow for variability in time, to account for the variability in the identification of the start and end times of an episode. As long as neighboring episodes nearly overlap in time (with a buffer of a single time bin on either end of each episode) and move forward in time, we concatenate them together into a single synchronous event.

For each event, we calculate the following information:

- i_{start}: starting neuron bin
- i_{end}: ending neuron bin
- t_{start}: starting time bin
- t_{end}: ending time bin
- E_{dir}: direction (up or down)
- E_{size}: event size (number of neuron bins covered).

The precise algorithm for concatenating episodes into events is as follows:

We begin with the list of episodes, which we sort first by start time, then by neuron bin. We concatenate episodes in consecutive neuron bins, allowing the event to travel in both directions (up and down) through the network. We consider events traveling up first, so we take the first episode in the sorted list, then search that list for an episode occurring in the next neuron bin. If the episodes occurred in overlapping (with buffer)

time bins, then we concatenate them. We only consider events that travel forward through time, though we include a time buffer of one time bin on either end. The time buffer is enforced, and we concatenate episodes, if either of the following conditions is met:

1. The episode in the next neuron bin starts within the time buffer of the current end time

2. The episode in the next neuron bin ends within the allowed time buffer of the current start time.

We repeat this process, searching the sorted list of episodes for an episode in the next neuron bin and concatenating, until we are no longer able to find an episode within the consecutive time and space constraints. After we have concatenated episodes traveling up through the network, we then repeat that process going down through the neuron bins. Finally, we include an event in our final list of events only if it covers at least 3 neuron bins (so if $E_{\text{size}} \geq 3$).

3.3.5 Simulation statistics

After we determine the synchronous events of a simulation, we then calculate a series of statistics. We calculate and record:

- event rate (number of synchonous events per second)
- event magnitude (number of neurons covered by events, proportional to the number of neurons in network and simulation time)
- inter-event intervals (IEI), an array of time between consecutive events
- IEI excess kurtosis and skew.

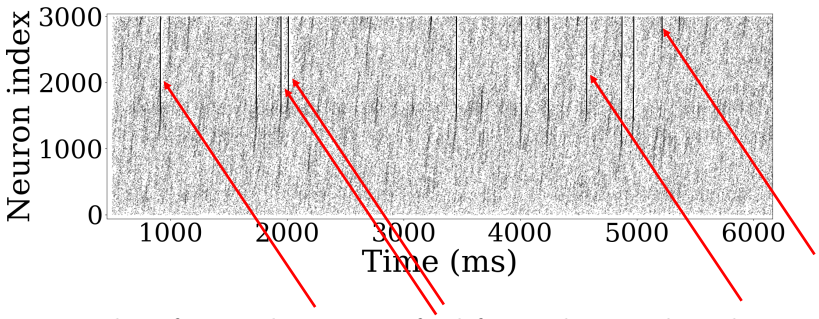


Figure 3.4: Raster plot of a simulation on a feed-forward network. When many neurons that are near each other fire at the same time, we see dark spots in the raster plot. Synchronous events correspond to dark lines that propagate through the network, some of which are indicated with arrows. Observe that in this feed-forward network, the events start in the middle of the network, travel forward through the network, and end at the top/end of the network.

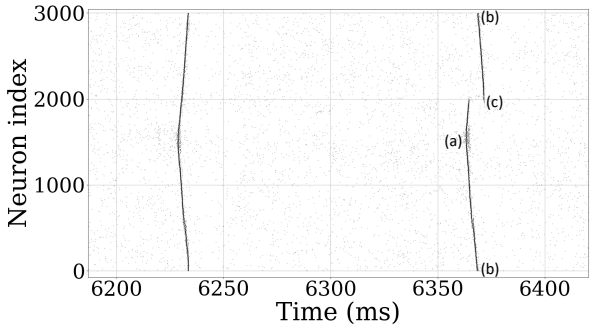


Figure 3.5: Zoomed-in raster plot of two events from a simulation of a recurrent network. In recurrent networks, we observe events that travel in both directions and wrap around. Here we have an event that starts at point (a), travels in both directions through the network, wraps around at point (b), and ends at point (c).

Chapter 4

Discussion of results

4.1 Network results

In this section, we demonstrate properties of the spatial SONET model. Following the work of Fuller [15], we extend the spatially homogeneous SONET model of Zhao et al. [1] to include dependence of connection probability on neuron location. The resulting model combines the microstructure of the SONET model (modulation of the motif frequencies of Figure 3.1) with a macrostructure (or global structure) that can be prescribed independently. We begin by illustrating the properties of the spatially homogeneous SONET model, and then demonstrate the SONET structure combined with simple one-dimensional geometries.

4.1.1 Homogeneous SONET model

Figure 4.1 shows how the modulation of reciprocal, convergent, divergent, and chain connection frequencies can be observed in the adjacency matrices of spatially homogeneous SONETs. Although not visible from the figure, with the added reciprocal connections in (b), this adjacency matrix is approximately symmetric about the main diagonal. With the added convergent connections in (c), we have a few postsynaptic neurons that receive connections from many presynaptic neurons. This results in a skewed distribution of the in-degree of the postsynaptic neurons, and is observed by horizontal striping in the adjacency matrix of the network. With the added divergent connections in (d), we have a few presynaptic neurons that connect to many postsynaptic neurons. This

results in a skewed distribution of the out-degree of the presynaptic neurons, and is observed by vertical striping in the adjacency matrix of the network. In (e) and (f) we observe both horizontal and vertical striping from increased convergent and divergent connections. The difference between (e) and (f) is the frequency of the chain motif. With reduced chain connections (e), those neurons with many incoming connections have few outgoing connections, and vis versa. On the other hand, with increased chain connections (f), the same neurons have many incoming and many outgoing connections, leading to symmetry in the striping.

4.1.2 Spatial SONET model

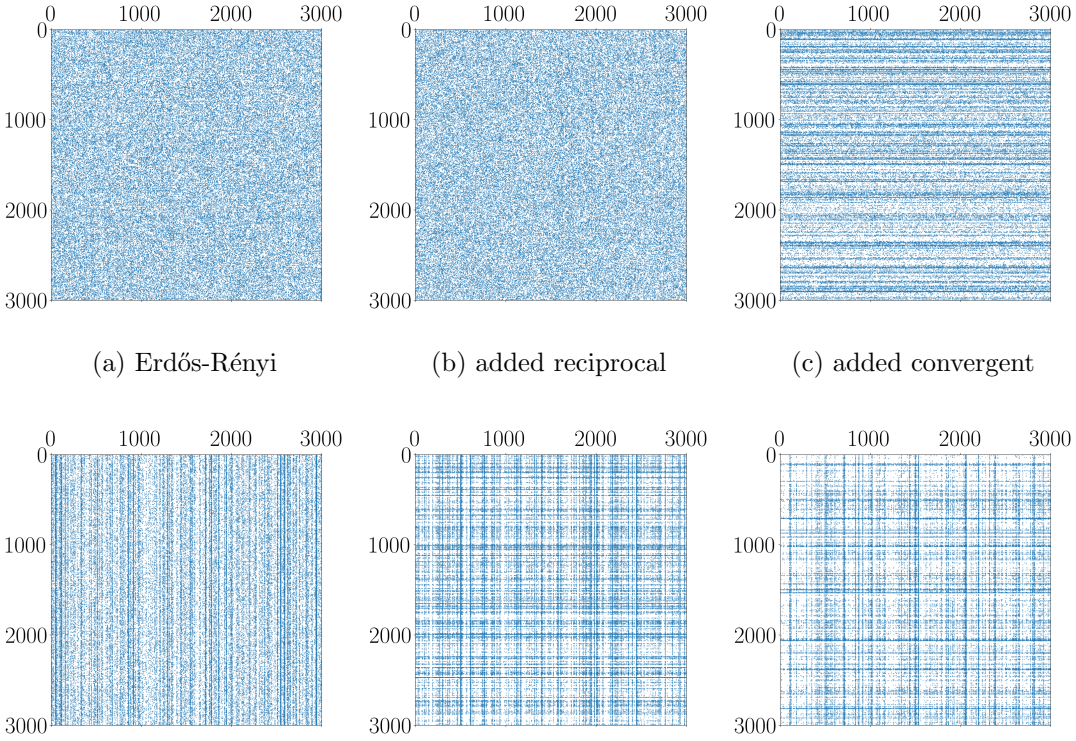
To introduce spatial structure to the SONET model, we considered the neurons as positioned along a one-dimensional lattice and set the connection probability to decay as a Gaussian function of the distance between neurons. Thus, the probability that neuron at position j connects to the neuron at position i is

$$p_{ij} = p_{\max} \exp\left(-d_{ij}^2/2\sigma^2\right),$$

where d_{ij} is the distance between neuron i and neuron j, and $p_{\max}$ is a scalar so the average number of connections per neuron is a prescribed value, p_{avg}. We consider various values of σ to adjust the spatial scale or "tightness" of the network. This is described with more detail in Chapter 3.

We examine two classes of network models that implement this spatial structure. The first is the recurrent model, where we suppose the neurons are arranged in a circle or ring, and let d_{ij} be the minimum distance (measured in neuron indices) between neurons i and j. The second class is feed-forward, where d_{ij} is the distance between neurons i and j if $i < j$ and is viewed as infinite otherwise. Our extension of the SONET model allows us to specify the second order motif frequencies (α values), which modulates the microstructure, while simultaneously prescribing the distance-dependence connection probability (p_{ij}), impacting the macrostructure.

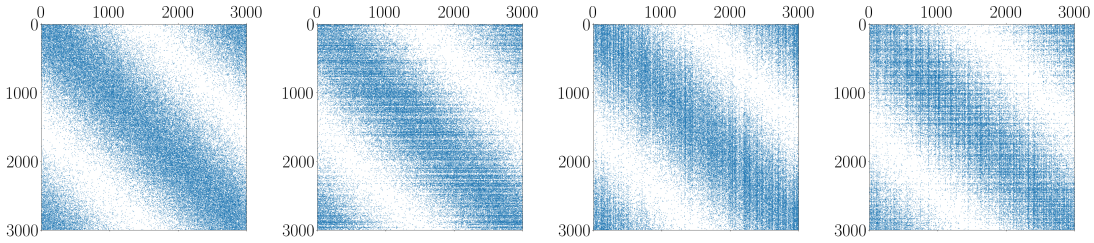
The algorithm developed for network generation does not effectively capture the reciprocal motif frequency. This is not a concern as Zhao et al. [1] demonstrated that reciprocal connections do not play a large role on the synchrony of the network. Further,



(a) Erdős-Rényi (b) added reciprocal (c) added convergent

(d) added divergent (e) reduced chain (f) added chain

Figure 4.1: Adjacency matrices of spatially homogenous networks with $N = 3000$ neurons, generated using $p = 0.01$ probability of connection. For each matrix, we have presynaptic neurons on the horizontal axis, and postsynaptic neurons on the vertical axis. **(a)** An Erdős-Rényi random network. Parameters: $\alpha_{\text{recip}} = 0$, $\alpha_{\text{conv}} = 0$, $\alpha_{\text{div}} = 0$, $\alpha_{\text{chain}} = 0$. **(b)** A random network with added reciprocal connections. Parameters: $\alpha_{\text{recip}} = 3$, $\alpha_{\text{conv}} = 0$, $\alpha_{\text{div}} = 0$, $\alpha_{\text{chain}} = 0$. **(c)** A random network with added convergent connections. Parameters: $\alpha_{\text{recip}} = 0$, $\alpha_{\text{conv}} = 3$, $\alpha_{\text{div}} = 0$, $\alpha_{\text{chain}} = 0$. **(d)** A random network with added divergent connections. Parameters: $\alpha_{\text{recip}} = 0$, $\alpha_{\text{conv}} = 0$, $\alpha_{\text{div}} = 3$, $\alpha_{\text{chain}} = 0$. **(e)** A random network with added convergent and divergent connections, and reduced chain connections. Parameters: $\alpha_{\text{recip}} = 0$, $\alpha_{\text{conv}} = 3$, $\alpha_{\text{div}} = 3$, $\alpha_{\text{chain}} = -0.9$. **(e)** A random network with added convergent, divergent, and chain connections. Parameters: $\alpha_{\text{recip}} = 0$, $\alpha_{\text{conv}} = 3$, $\alpha_{\text{div}} = 3$, $\alpha_{\text{chain}} = 3$.



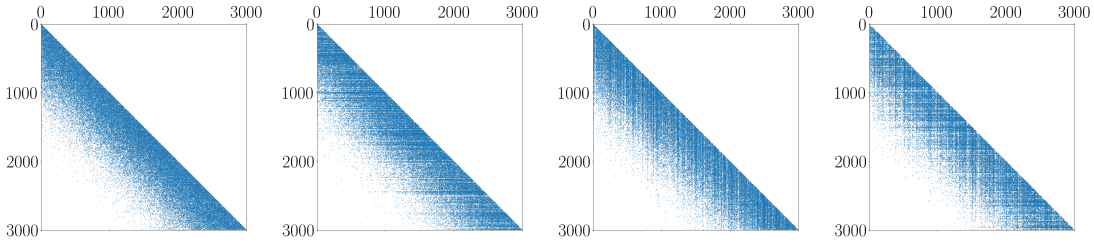
(a) recurrent (b) added convergent (c) added divergent (d) added chain

Figure 4.2: Adjacency matrices of recurrent networks with $N = 3000$ neurons, generated using $p = 0.01$ probability of connection, and a tightness factor of $\sigma = 500$. As with figure 4.1, presynaptic neurons are on the horizontal axis, and postsynaptic neurons are on the vertical axis. **(a)** A recurrent network with independent connections. Parameters: $\alpha_{\text{recip}} = 0, \alpha_{\text{conv}} = 0, \alpha_{\text{div}} = 0, \alpha_{\text{chain}} = 0$. **(b)** A recurrent network with added convergent connections. Parameters: $\alpha_{\text{recip}} = 0, \alpha_{\text{conv}} = 1, \alpha_{\text{div}} = 0, \alpha_{\text{chain}} = 0$. **(c)** A recurrent network with added divergent connections. Parameters: $\alpha_{\text{recip}} = 0, \alpha_{\text{conv}} = 0, \alpha_{\text{div}} = 1, \alpha_{\text{chain}} = 0$. **(d)** A recurrent network with added chain connections. Parameters: $\alpha_{\text{recip}} = 0, \alpha_{\text{conv}} = 0.9, \alpha_{\text{div}} = 0.9, \alpha_{\text{chain}} = 0.9$.

as will later be discussed, we primarily focus on feed-forward networks, where reciprocal connections are not present. Hence, we ignore the reciprocal motif frequency and consider networks only with varying convergent, divergent, and chain motif frequencies.

In Figures 4.2 and 4.3 we observe the effect of convergent, divergent, and chain connections in recurrent and feed-forward networks, which appear as horizontal and vertical striping, and symmetry, respectively, in the adjacency matrices. The striping demonstrates that we have successfully captured the SONET microstructure of Figure 4.1. The spatial macrostructure is evident from the larger scale pattern, with the dots concentrated on the diagonal and corners for the recurrent model (Figure 4.2), and concentrated below the diagonal for the feed-forward case (Figure 4.3).

Zhao et al. [1] demonstrated the relationship between the SONET parameters (α values) and the network degree distribution. Figure 4.4 illustrates how this relationship is still present in the spatial model. Increased convergent connections lead to a skew in the in-degree distribution, which is caused by the presence of a few nodes with in-degree much larger than average (Figure 4.4b). Similarly, increased divergent connections lead to a skew in the out-degree distribution (Figure 4.4c). The effect of the chain connections on the relationship between in-degree and out-degree distributions is shown in Figures



(a) feed-forward (b) added convergent (c) added divergent (d) added chain

Figure 4.3: Adjacency matrices of feed-forward networks with $N = 3000$ neurons, generated using $p = 0.01$ probability of connection, and a tightness factor of $\sigma = 500$. As with figure 4.1 and 4.2, presynaptic neurons are on the horizontal axis, and postsynaptic neurons are on the vertical axis. **(a)** A feed-forward network with independent connections. Parameters: $\alpha_{\text{recip}} = 0, \alpha_{\text{conv}} = 0, \alpha_{\text{div}} = 0, \alpha_{\text{chain}} = 0$. **(b)** A feed-forward network with added convergent connections. Parameters: $\alpha_{\text{recip}} = 0, \alpha_{\text{conv}} = 1, \alpha_{\text{div}} = 0, \alpha_{\text{chain}} = 0$. **(c)** A feed-forward network with added divergent connections. Parameters: $\alpha_{\text{recip}} = 0, \alpha_{\text{conv}} = 0, \alpha_{\text{div}} = 1, \alpha_{\text{chain}} = 0$. **(d)** A feed-forward network with added chain connections. Parameters: $\alpha_{\text{recip}} = 0, \alpha_{\text{conv}} = 0.9, \alpha_{\text{div}} = 0.9, \alpha_{\text{chain}} = 0.9$.

4.4d and 4.4e. Only with increased chain connections are there nodes with both large in-degree and out-degree.

In figure 4.5 we observe the effect of various σ values on the spatial tightness of recurrent and feed-forward networks. We can see that as σ decreases the network becomes tighter. In recurrent networks, this spatial tightness is about the main diagonal and corners of the adjacency matrix. In feed-forward networks, we observe the spatial tightness below the main diagonal of the adjacency matrices.

4.2 Dynamical regimes

By altering parameters of the Poisson process of external input in our simulation, we have discovered two fundamental dynamical regimes. When the magnitude of external input is low, but the rate of external input is high, we observe events that occur frequently, with low variance of the time between events (inter-event interval or IEI); we call this the regular regime. However, when the magnitude of external input is high, but the rate of external input is low, events occur more randomly, with high variance IEI distribution; we call this the irregular regime.

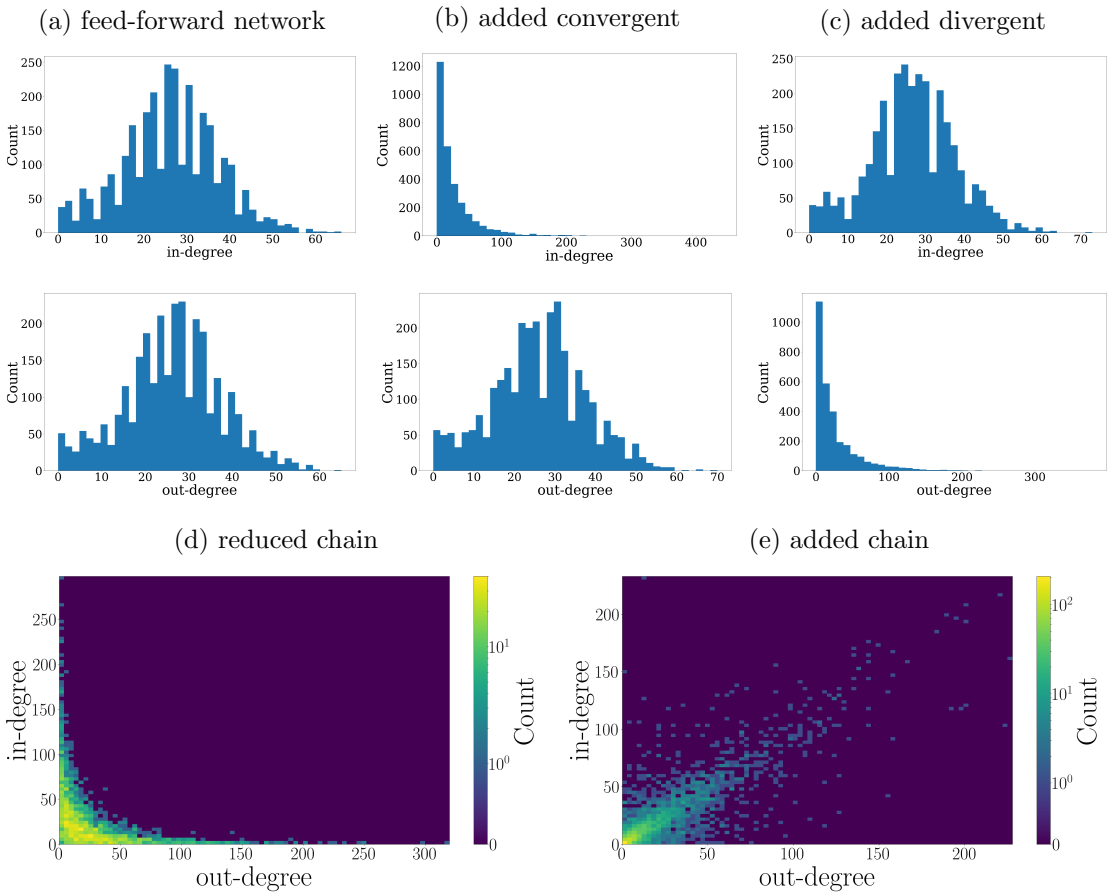
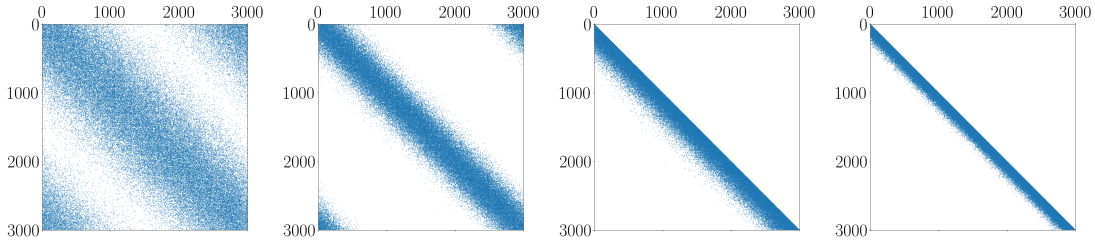


Figure 4.4: Histograms comparing the distributions of in-degree and out-degree of networks with varying convergent and divergent connections. Each network was generated with $N = 3000$ neurons using $p = 0.01$ probability of connection (about 30 connections per neuron), with feed-forward macrostructure with $\sigma = 500$. **(a)** Histograms of in- and out-degree distributions of an independent network. Parameters: $\alpha_{\text{recip}} = 0, \alpha_{\text{conv}} = 0, \alpha_{\text{div}} = 0, \alpha_{\text{chain}} = 0$. The adjacency matrix for this network can be seen in figure 4.3a. **(b)** Histograms of in- and out-degree distributions of a network with added convergent connections. Parameters: $\alpha_{\text{recip}} = 0, \alpha_{\text{conv}} = 1, \alpha_{\text{div}} = 0, \alpha_{\text{chain}} = 0$. The adjacency matrix for this network can be seen in figure 4.3b. **(c)** Histograms of in- and out-degree distributions of a network with added divergent connections. Parameters: $\alpha_{\text{recip}} = 0, \alpha_{\text{conv}} = 0, \alpha_{\text{div}} = 1, \alpha_{\text{chain}} = 0$. The adjacency matrix for this network can be seen in figure 4.3c. **(d)** Histogram of in- and out-degree distributions of network with reduced chain connections. Parameters: $\alpha_{\text{recip}} = 0, \alpha_{\text{conv}} = 0.9, \alpha_{\text{div}} = 0.9, \alpha_{\text{chain}} = -0.9$. **(e)** Histogram of in- and out-degree distributions of network with added chain connections. Parameters: $\alpha_{\text{recip}} = 0, \alpha_{\text{conv}} = 0.9, \alpha_{\text{div}} = 0.9, \alpha_{\text{chain}} = 0.9$. The adjacency matrix for this network can be seen in figure 4.3d.



(a) $\sigma = 500$ (b) $\sigma = 200$ (c) $\sigma = 200$ (d) $\sigma = 100$

Figure 4.5: Adjacency matrices of networks with $N = 3000$ neurons, generated using $p = 0.01$ probability of connection. For each matrix, we have presynaptic neurons on the horizontal axis, and postsynaptic neurons on the vertical axis. Also, there is no imposed spatial structure on these networks, so we have $\alpha_{\text{recip}} = 0, \alpha_{\text{conv}} = 0, \alpha_{\text{div}} = 0$, and $\alpha_{\text{chain}} = 0$. **(a)** and **(b)** are recurrent networks with $\sigma = 500$ and $\sigma = 200$, respectively. **(c)** and **(d)** are feed-forward networks with $\sigma = 200$ and $\sigma = 100$, respectively.

We observed that the precise values of the rate and magnitude of external input for each regime varies based on the macrostructure of the network. Hence, to better identify these regimes we generated ten independent (all α values set to zero) networks of each macrostructure type: homogeneous, recurrent with $\sigma = 50$, and feed-forward with $\sigma = 100$. Then we ran simulations on each network and averaged the event rate and IEI skew over the ten simulations of each network type. We chose Poisson process parameters for the regular regime so that the mean event rate was greater than 30 events per second and the mean IEI skew was less than 0.5 in absolute value. For the irregular regime, we chose parameters so that the mean event rate was less than 10 events per second and the mean IEI skew was greater than 1. Table 4.1 gives the external input magnitude (I^{ext}) and external input rate (λ) parameters chosen for each macrostructure type and regime.

Raster plots that illustrate the dynamical regimes are given in Figure 4.6. We observe that the nature of events in the irregular regime is fundamentally stochastic, which is not the case in the regular regime.

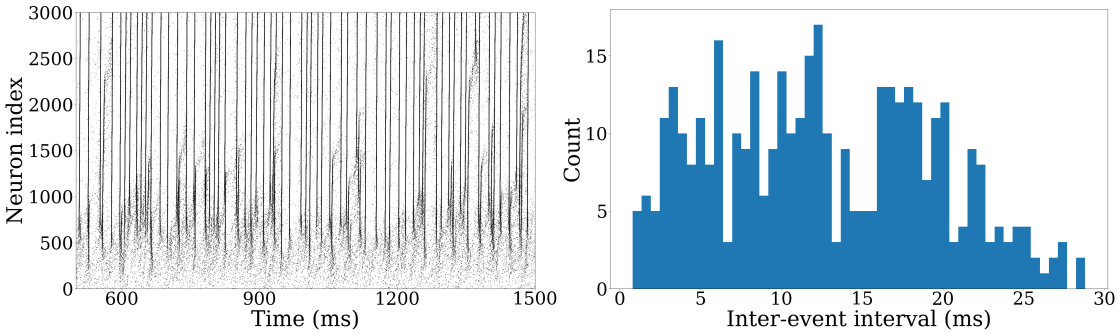
4.3 Simulation results

We used the Brian spiking neuronal network simulator [17] to simulate a leaky integrate-and-fire model on networks of $N = 3000$ excitatory neurons with instantaneous synapses.

Macrostructure	Regular regime		Irregular regime	
	I^{ext}	λ	I^{ext}	λ
Homogeneous	1 mv	250 Hz	1.65 mV	110 Hz
Recurrent	1 mv	250 Hz	1.5 mV	113 Hz
Feed-forward	1 mv	250 Hz	1.5 mV	160 Hz

Table 4.1: Dynamical regime parameters by macrostructure type

(a) Regular regime: low input magnitude, high input rate



(b) Irregular regime: high input magnitude, low input rate

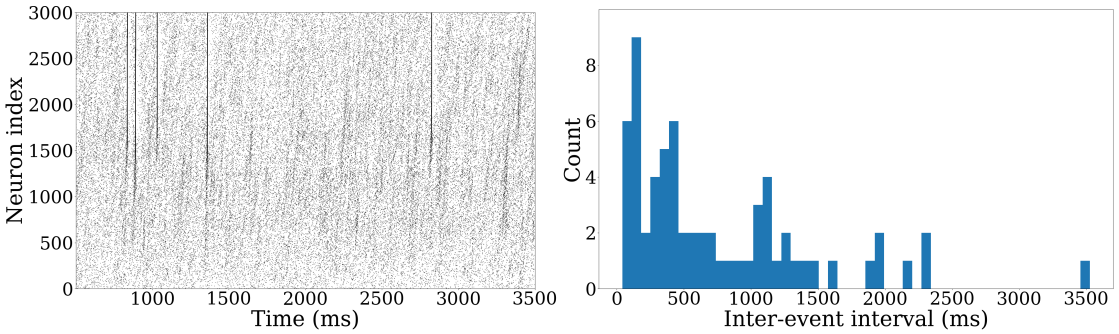


Figure 4.6: Raster plots and histograms of the inter-event intervals (IEIs) of two different simulations of the same feed-forward network. Each left panel shows a raster plot of a single simulation and the right panel is a histogram of the inter-event intervals (time between events) for that simulation. Note the difference of the temporal scales across regime. **(a)** Regular regime simulation. Event rate: 78.8 events per second, IEI skew: 0.1444.**(b)** Irregular regime simulation. event rate was 1.32 events per second and the IEI skew was 1.5495.

We modulate the dynamical regime by adjusting the external input magnitude (I^{ext}) and external input rate (λ). Then we apply our event-detection algorithm to identify synchronous events. We use the event rate as our measure of network synchrony.

Zhao et al. [1] saw no influence of divergence motif on synchrony, and Nykamp et al. [11] showed how the mean-field equations are not influenced by divergence. However, Liu and Nykamp [12] found common input (similar to the divergence motif) influenced synchrony in layered feed-forward networks, though Rosenbaum et al. [13, 14] showed overlap (similar to common input and divergent connections) played a smaller role than pooling (similar to the convergent connections) in determining correlation in layered feed-forward networks.

Unfortunately, it is difficult to compare the results of Zhao et al. to others, as they used a different network model. However, these results seem to point to different network effects from feed-forward networks vs. recurrent networks. Now that we have a spatial SONET model, we can directly test this hypothesis, since we can create networks with similar microstructures (second order motif frequencies) whose only difference is macrostructure (feed-forward vs. recurrent vs. spatially homogeneous).

4.3.1 Role of divergence in feed-forward networks

The first thing we wanted to test with spatial SONETs is whether or not the divergence motif has an influence on synchrony (event rate) in feed-forward models. So, we generated 300 feed-forward networks where we varied α_{div} (let α_{div} be chosen uniformly at random from $[0, 0.5]$) and kept all other α parameters at 0. For each network, we ran simulations in both regular and irregular dynamical regimes, and calculated the event rate as a measure of synchrony.

When we plotted event rate vs. $\hat{\alpha}_{\text{div}}$ (recall from section 3.1.2 that we use $\hat{\alpha}_{\text{div}}$ as an approximation of α_{div}) for simulations in the regular dynamical regime, we did not see any increase in synchrony as a function of $\hat{\alpha}_{\text{div}}$ (see Figure 4.7a). Although the range of the event rate did increase as a function of $\hat{\alpha}_{\text{div}}$, we did not observe a substantial positive correlation between divergent motif frequency and event rate. Hence, even with feed-forward networks, we matched the previous observations made by Zhao et al. for spatially homogeneous networks. However, the simulations of Figure 4.7a were all in the regular dynamical regime. This may put the network in a state that could be described

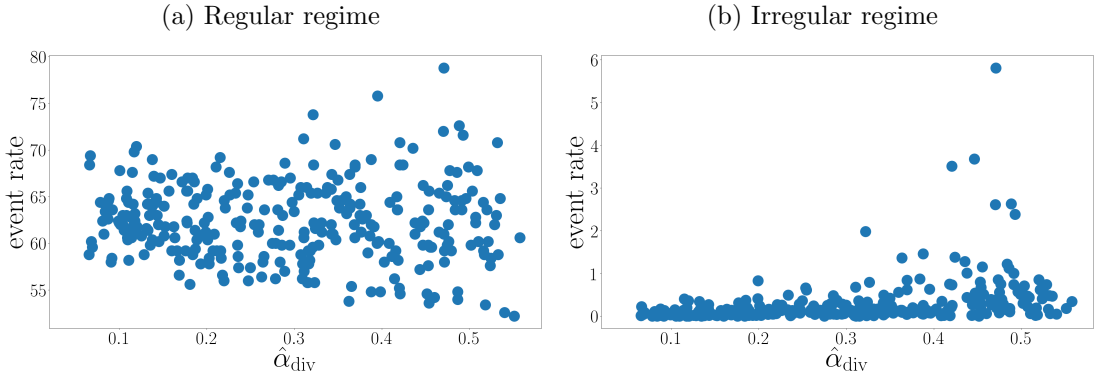


Figure 4.7: Scatter plots of the event rate vs. $\hat{\alpha}_{\text{div}}$ for 300 feed-forward network simulations in our two dynamical regimes with varying divergent motif frequencies. Each dot corresponds to the event rate of one network simulation. For each network, α_{div} was sampled uniformly and independently from $[0, 0.5]$, all other α values were zero (i.e., $\alpha_{\text{recip}} = \alpha_{\text{conv}} = \alpha_{\text{chain}} = 0$). Note the different scales of event rate in each regime. Regular regime simulation: Poisson input parameters: magnitude $I^{\text{ext}} = 1\text{mV}$, rate $\lambda = 250$ Hz. Simulation length: 5 seconds. Irregular regime simulation: Poisson input parameters: $I^{\text{ext}} = 1.5\text{mV}$, $\lambda = 116$ Hz. Simulation length: 100 seconds.

by the mean-field equations, as observed by Nykamp et al. [11], in which divergence plays no role.

We hypothesized that the frequency of divergent connections might play a bigger role if we put the network into an irregular regime that is more fundamentally stochastic. In such a regime, mean-field equations clearly wouldn't describe the network behavior, and the influence of divergence may be revealed. Indeed, as shown in Figure 4.7b, when we increase the external input magnitude while decreasing the external input rate, the resulting irregular regime simulations show a positive correlation between $\hat{\alpha}_{\text{div}}$ and synchrony as measured by event rate.

Thus, at least for feed-forward networks, the influence of the divergent motif on synchrony depends on the dynamical regime of the network. Only when the dynamics appear to be fundamentally stochastic does increasing the relative frequency of the divergent motif increase synchrony, as measured by the rate of synchronous events.

4.3.2 Role of divergence in recurrent networks

The results of Figure 4.7 show that synchrony of feed-forward networks in an irregular dynamical regime is influenced by the frequency of divergent connections. A natural question is whether such influence of divergence is limited to feed-forward networks, or if it can also be observed in recurrent networks given a stochastic regime. Although mean-field models of recurrent networks that show no dependence on divergent motif frequency could be developed, as Nykamp et al. [11] did for spatially homogeneous networks, if we can put the network into a dynamical regime where the mean-field dynamics do not apply, it may be possible to uncover an influence of α_{div}.

Certainly, Figure 4.8b shows that one can obtain a positive correlation between $\hat{\alpha}_{\text{div}}$ and event rate when simulations are shifted to the irregular dynamical regime. In this regime, the synchronous events are far from periodic, so they cannot be explained by a periodic orbit emanating from a Hopf bifurcation in the mean-field dynamics. If we change the simulation parameters to the more typical regular dynamical regime where the synchronous events are more periodic, the correlation of event rate with $\hat{\alpha}_{\text{div}}$ disappears, yielding a result consistent with predictions from mean-field dynamics (see Figure 4.8a).

4.3.3 Role of divergence in spatially homogeneous networks

Our results indicate that the key feature underlying dependence on divergence is not the macrostructure (feed-forward or recurrent). Instead, the effect of divergence on synchrony is revealed for both spatial macrostructures only when the network is put into a regime where the synchronous events appear to be generated by a stochastic mechanism (the irregular dynamical regime). As a final test of the hypothesis that the macrostructure does not play a critical role, we went back to the spatially homogeneous SONET model of Zhao et al. [1], where the probability of a connection between any pair of neurons does not depend on neuron location.

As shown in Figure 4.9, we obtain the same results for spatially homogeneous SONETs: we observe a correlation between $\hat{\alpha}_{\text{div}}$ and event rate in the irregular regime which disappears in the regular regime. We confirm the Zhao et al. results of limited

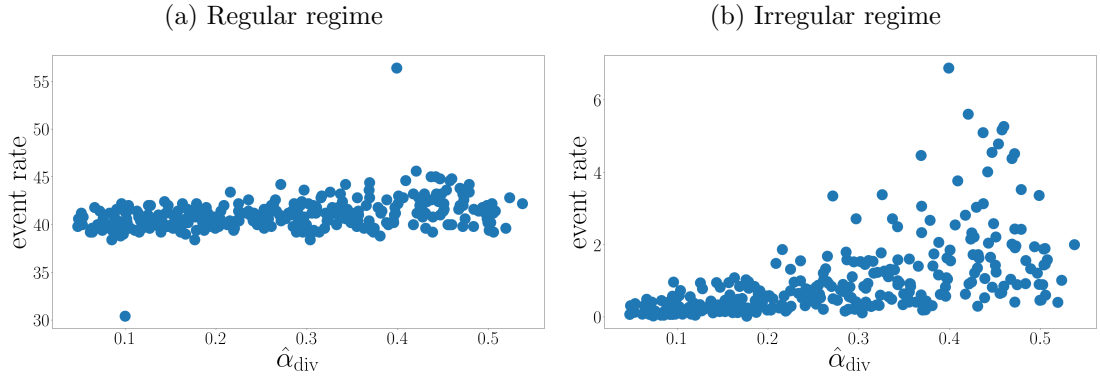


Figure 4.8: Scatter plots of the event rate vs. $\hat{\alpha}_{\text{div}}$ for 300 recurrent network simulations in our two dynamical regimes with varying divergence motif frequencies. Each dot corresponds to the event rate of one network simulation. For each network, α_{div} was sampled uniformly from $[0, 0.5]$, all other α values were zero (i.e., $\alpha_{\text{recip}} = \alpha_{\text{conv}} = \alpha_{\text{chain}} = 0$). Regular regime simulation: Poisson input parameters: $I^{\text{ext}} = 1\text{mV}$, $\lambda = 250$ Hz. Simulation length: 5 seconds. Irregular regime simulation: Poisson input parameters: $I^{\text{ext}} = 1.5\text{mV}$, $\lambda = 113$ Hz. Simulation length: 100 seconds.

effect of divergence motif frequency (although we are using a different measure for synchrony) for the case of periodic synchronous events, and we reveal that those results do not apply when stochasticity becomes evident.

4.3.4 Role of convergence

Rosenbaum et al. [13][14] discovered that pooling (similar to convergence) dominates the correlation of network activity when both pooling (convergence) and overlap (similar to divergence) are present in a layered feed-forward network. However, it is important to note that their definition of pooling (convergence) involves the mean in-degree, which is a first order statistic, whereas our definition of convergence depends on the variance of in-degree, a second-order statistic. Furthermore, Zhao et al. [1] observed the opposite effect, where synchrony decreased with convergence.

We explore the effect of α_{conv} by generating 300 networks of each macrostructure type (feed-forward, recurrent, and spatially homogeneous) with α_{div} and α_{conv} each sampled uniformly at random from $[0, 0.5]$. Our results are consistent with those of Rosenbaum et al. in that convergent motif frequency had a much larger effect than divergent motif frequency on event rate in feed-forward netwoks. Interestingly, the

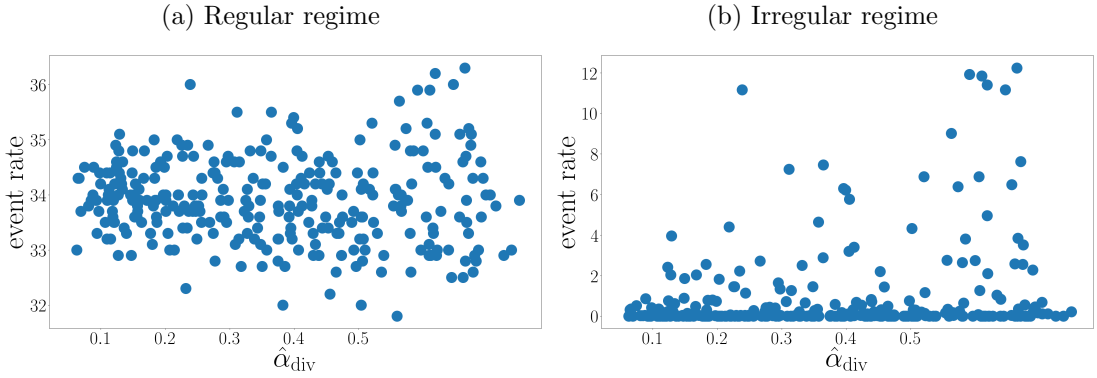


Figure 4.9: Scatter plots of the event rate vs. $\hat{\alpha}_{\text{div}}$ for 300 spatially homogeneous network simulations in our two dynamical regimes with varying divergent motif frequencies. Each dot corresponds to the event rate of one network simulation. For each network, α_{div} was sampled uniformly from $[0, 0.5]$, all other α values were zero (i.e., $\alpha_{\text{recip}} = \alpha_{\text{conv}} = \alpha_{\text{chain}} = 0$). Regular regime simulation: Poisson input parameters: $I^{\text{ext}} = 1\text{mV}$, $\lambda = 250$ Hz. Simulation length: 5 seconds. Irregular regime simulation: Poisson input parameters: $I^{\text{ext}} = 1.65\text{mV}$, $\lambda = 110$ Hz. Simulation length: 50 seconds.

strong effect of α_{conv} was present for both regular and irregular regimes (see Figure 4.10).

To investigate if this effect of α_{conv} depends on network macrostructure, we repeated the simulations with a recurrent spatial macrostructure, and the spatially homogeneous SONETs. As shown in Figures B.1 and B.2, we observed the same dependence of event rate on frequency of convergent motif for both network types, in both regular and irregular dynamical regimes: synchrony, as measured by event rate, increased with $\hat{\alpha}_{\text{conv}}$, and $\hat{\alpha}_{\text{conv}}$ had a stronger effect than $\hat{\alpha}_{\text{div}}$.

This observation of spatially homogeneous networks leaves us wondering how to explain the results of Zhao et al. One theory is that perhaps the synchrony is not exactly periodic, so the Kuramoto order parameter may not be able to detect an increase in synchrony as our event-detection algorithm does.

4.3.5 Role of chain connections

Zhao et al. [1] noticed that the motif with the strongest effect on synchrony in was the chain motif, and its primary influence was shown in the mean-field analysis of Nykamp et al. [11]. We investigated if the chain motif would still strongly influence synchrony even

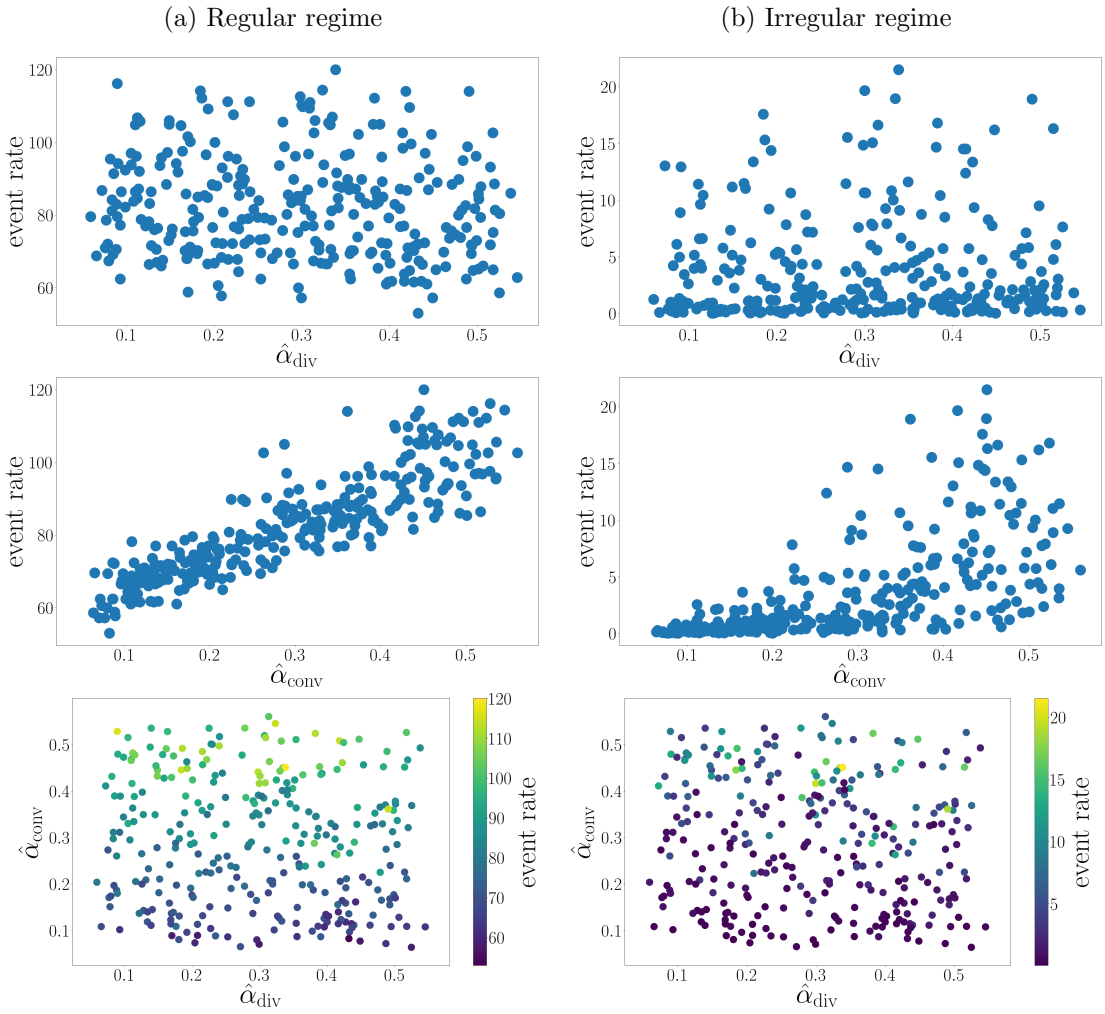


Figure 4.10: Scatter plots of results from simulations of 300 feed-forward networks with varying convergent and divergent motif frequencies. Each dot represents one simulation. The left column contains plots from simulations done in the regular dynamical regime, and the right plots are from the irregular regime. The top plot is event rate vs. $\hat{\alpha}_{\text{div}}$, the middle plot is event rate vs. α_{conv}, and the bottom plot is $\hat{\alpha}_{\text{conv}}$ vs. $\hat{\alpha}_{\text{div}}$ with the color of the dots corresponding to the event rate. Regular regime simulation: Poisson input parameters: $I^{\text{ext}} = 1\text{mV}$, $\lambda = 250$ Hz. Simulation length: 5 seconds. Irregular regime simulation: Poisson input parameters: $I^{\text{ext}} = 1.5\text{mV}$, $\lambda = 116$ Hz. Simulation length: 100 seconds.

in cases where we do not have the mean-field results. We generated 300 networks with α_{conv} and α_{div} sampled uniformly at random from [0.0.5], and α_{chain} sampled uniformly at random from $[-0.5, 0.5]$.

Results from feed-forward networks simulated in the irregular regime are shown in Figure 4.11, where we see that the effect of the chain motif supersedes the effect of convergent and divergent motifs. The top right plot of the Figure 4.11 illustrates that the event rate increases with $\hat{\alpha}_{\text{chain}}$, but very little (if at all) with $\hat{\alpha}_{\text{div}}$.

We also performed simulations on feed-forward networks in the regular regime (see Figure B.3), as well as recurrent and spatially homogeneous networks in both regular and irregular dynamical regimes (see Figures B.4, B.5, B.6, and B.7). Thus, we see that, regardless of macrostructure and dynamical regime, α_{chain} plays the primary role in impacting the event rate, with α_{conv} a secondary role. Furthermore, any role of α_{div} is not evident in these plots, as it is dwarfed by the effects of α_{chain} and α_{conv}.

4.3.6 Errors with event detection algorithm

One drawback of basing our synchrony measure on the ability to identify events is that, under some circumstances, it is challenging to detect events, or even clearly define what should be considered an event. Once event initiation becomes extremely frequent, they are likely to run into each other before they have the opportunity to propagate far through the network. In these cases, our event detection algorithm cannot identify many events, and so is unable to calculate an accurate event rate. In this situation, our algorithm either inappropriately concatenates individual events, or misses the events all together because each event does not span enough neuron bins for our algorithm to consider it an event.

An example of a case where events are initiated too rapidly to be detected is shown in Figure 4.12. The raster plot displays many events, which should mean a very high event rate. Although, our event detection algorithm calculates an event rate of 103.2 events per second, the raster plot indicates the actual event rate may be much higher. Many events begin, but then end very quickly. In most situations the events are short because they run into (refractory periods of) other events. We conclude that there are many events that we can see, but are not detected by our algorithm due to how short they are (they do not cover enough neuron bins for our algorithm to consider them

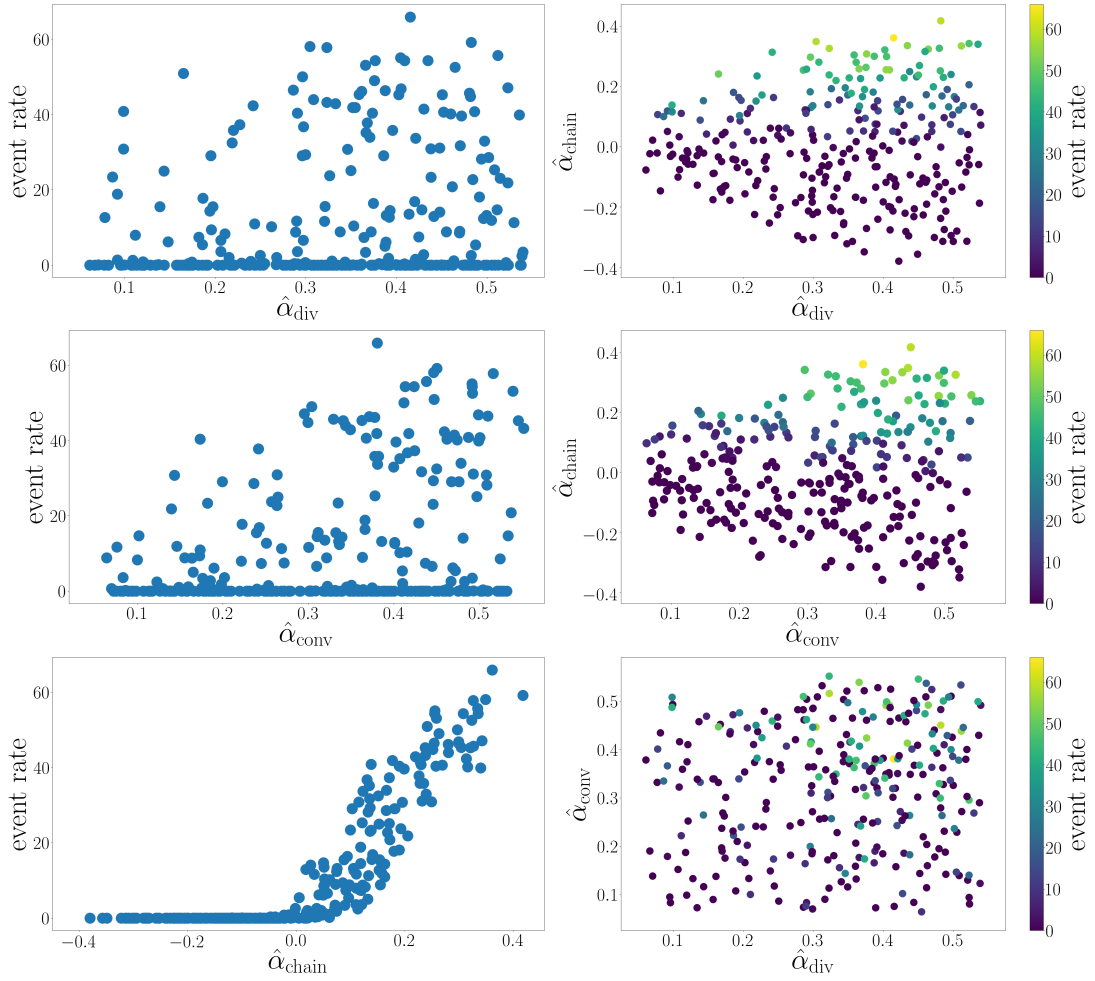


Figure 4.11: Scatter plots of results from 300 feed-forward networks with varying convergent, divergent, and chain connections in the irregular regime. Each dot represents one simulation. For each network α_{div} and α_{conv} were sampled uniformly from $[0, 0.5]$, α_{chain} was sampled uniformly from $[-0.5, 0.5]$, and α_{recip} was zero. In the left column we have event rate plotted against the three varying α values. The right column plots the different α values against each other with the color of the dots corresponding to the event rate. Poisson input parameters: $I^{\text{ext}} = 1.5\text{mV}$, $\lambda = 116$ Hz. Simulation length: 100 seconds.

events).

Figure 4.13a illustrates how the poor detection of high event rates can influence the shape of the event rate scatter plot. Here, we notice that the event rate increases with $\hat{\alpha}_{\text{conv}}$ until $\hat{\alpha}_{\text{conv}}$ is approximately 0.4, then decreases steadily. This decrease is due to the failure to detect events. (Figure 4.12 is one of these such simulations with $\hat{\alpha}_{\text{conv}} \approx 0.5$ but with a calculated event rate of 103.2 events per second.)

To trust our results, it is important to be able to identify when the event-detection algorithm is failing to detect events. Otherwise, there is no way to know from plots if a lower event rate indicates low synchrony or extremely high synchrony. The difference can be distinguished by observing the event size (number of neuron bins covered by the event).

In Figure 4.13b we see results from regular regime simulations on recurrent networks, where the event size decreases with $\hat{\alpha}_{\text{conv}}$ and appears to plateau as $\hat{\alpha}_{\text{conv}}$ approaches and exceeds 0.5. Furthermore, we observe that this plateau is very low, less than five neuron bins on average. Since we require all events to cover at least three neuron bins, an average event size close to three can be used as an indicator that our event-detection algorithm is failing. Furthermore, since the average event size is large for many networks with $\hat{\alpha}_{\text{conv}}$ less than 0.3, we presume that the low event rates for those networks are valid.

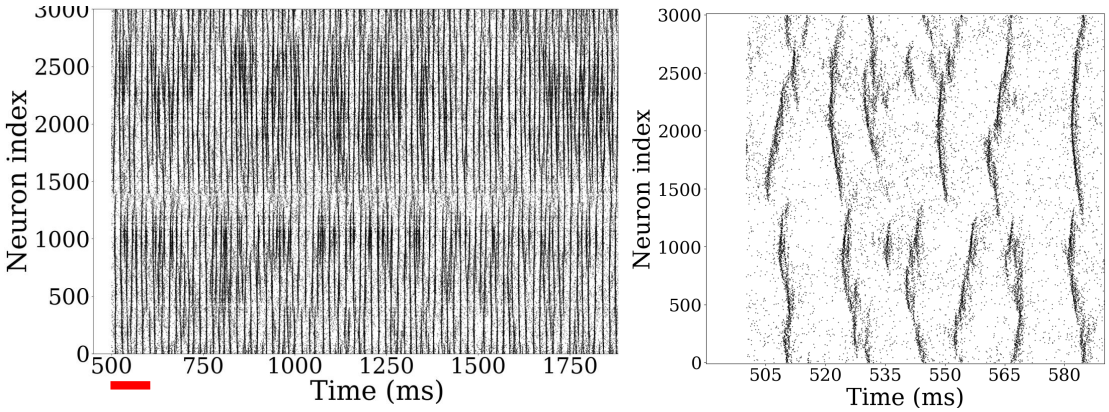


Figure 4.12: A raster plots of one regular regime simulation of a recurrent network with $\alpha_{\text{recip}} = 0, \alpha_{\text{conv}} \approx \alpha_{\text{div}} \approx 0.5$, and $\alpha_{\text{chain}} \approx 0$. The plot on the right is a portion of the plot on the left, indicated by a small bar from 500 ms to about 600 ms under the plot on the left. Poisson input parameters: $I^{\text{ext}} = 1\text{mV}$, $\lambda = 250$ Hz. Simulation length: 5 seconds

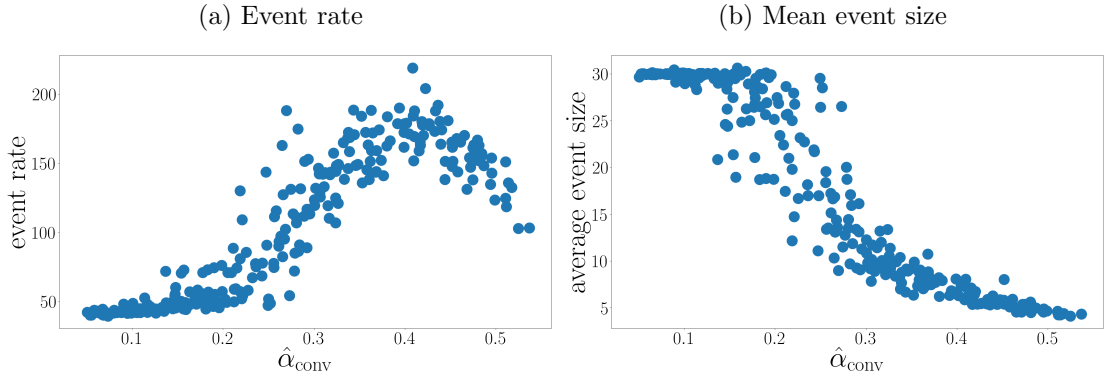


Figure 4.13: Scatter plots of event rate and average event size vs. $\hat{\alpha}_{\text{conv}}$ from 300 recurrent networks with varying divergent and convergent connection frequencies in the regular regime. Each dot represents one simulation. On the left we have event rate vs. $\hat{\alpha}_{\text{conv}}$, and on the right we see average event size vs. $\hat{\alpha}_{\text{conv}}$ For each network α_{div} and α_{conv} were sampled uniformly from $[0, 0.5]$ and the other α values were zero (i.e., $\alpha_{\text{recip}} = \alpha_{\text{chain}} = 0$). Poisson input parameters: $I^{\text{ext}} = 1\text{mV}$, $\lambda = 250$ Hz. Simulation length: 5 seconds.

Chapter 5

Addendum: Stochastic simulations

Our results indicate an interesting effect: that the influence of network structure on synchrony is dependent on dynamical regime. Since the irregular dynamical regime appears to be fundamentally stochastic, it leads us to question whether we can capture the effect of divergent motif frequency on the event rate with a simple stochastic model.

Our goal was to then analyze the stochastic model to initiate an understanding of the mechanism by which divergence modulates synchrony in integrate-and-fire simulations. However, we discovered that the simple stochastic model did not exhibit the dependence on divergent motif frequency that we had previously observed. We conclude that our simple stochastic model is missing key features of the integrate-and-fire model, which gives us some insight into the nature of the mechanism.

5.1 Stochastic simulation

We implement a simple stochastic simulation using a contact process on a directed graph. Our network generation algorithm produces a directed graph as a sparse adjacency matrix, so we will use this for our stochastic simulation.

5.1.1 Stochastic model

We consider a contact process on a directed graph. Each neuron can be active/firing (state 1) or not (state 0) and we define a continuous-time Markov process with state space $\{0,1\}^N$, where N is the number of neurons in the network. We assume that neuron i transitions from active state 1 to static/inert state 0 at a constant rate $\mu = 1$, and from static state 0 to active state 1 at a rate depending on external input and the state of neurons that connect to i:

$$\lambda_i = \lambda^{\text{ext}} + \lambda^* \sum_j W_{ij}\xi_j,$$

where λ^{ext} is the rate of external input, λ^* is the connectivity strength of the simulation, W is the adjacency matrix for the network, and ξ_j is the state of neuron j.

To simulate this process, we let

$$\lambda^{\text{total}} = \sum_i \Big(\lambda_i(1-\xi_i) + \mu\,\xi_i\Big).$$

Then we generate an exponential random variable T_n at rate λ^{total} to represent the time of the next neural state switch. Therefore, the kth neural state switch occurs at time $t = \sum_{n=1}^{k} T_n$. The neuron that switches states is neuron i with probability

$$\left(\mu\xi_i + \lambda_i(1-\xi_i)\right)/\lambda^{\text{total}}.$$

5.1.2 Simulation length

There are many features of spiking neuron simulations that our stochastic simulation is unable to capture. Most notably, we do not take the refractory period into account. Therefore, we found that as soon as the simulation reaches a state where many of the neurons in the network are active, it tends to remain in that state of high neural activity for the duration of the simulation.

We found that 25000 state switches was enough to get to the point where the network became active. Hence, we run each simulation for 25000 state switches, then calculate how long it took for the network to become active, and recorded this as the event time.

We repeat this simulation procedure multiple times then consider the distribution of the event times, which should be analogous to the inter-event intervals (IEI) of a spiking neuron simulation using the leaky integrate-and-fire model.

5.1.3 Event time calculation

We calculate the event time to be the first time when the number of active neurons exceeds a threshold. We set the threshold to be half of the average number of neurons active for the last 20% of state switches (the last 5000 switches). Hence, we first calculate the time at which 80% of state switches have occurred, then average the number of active neurons over time for the rest of the simulation (the remaining 5000 state switches). We consider this value to be a plateau of the number of active neurons, and set the event threshold to be half of the plateau. In Figure 5.1 we see a raster plot of a single 25000-state-switch stochastic simulation, and a plot of the number of active neurons as a function of time with the plateau and event threshold for the same simulation.

5.1.4 Networks used and simulation results

When we ran spiking neuron simulations using a leaky integrate-and-fire model, we observed that divergent connection motif frequency may have an effect on the event rate in the irregular regime. Since we were able to observe this phenomenon for spatially homogeneous networks (the macrostructure with the least amount of prescribed structure), we decided to test our stochastic model on the same.

We ran 200 individual 25000-state-switch stochastic simulations on each network. We consider the event time (time until the number of active neurons reaches threshold) for each individual simulation to be analogous to the inter-event-interval from integrate-and-fire simulations. Thus, we calculated an event rate by dividing 200 (the number of simulations, and thus, the number of events) by the sum of the event times for all 200 simulations on that network. In Figure 5.2, we see that $\hat{\alpha}_{\text{div}}$ had the reverse effect on event rate than what was anticipated.

5.1.5 Hypotheses of missing features

Since we discovered that the stochastic model did not exhibit the dependence on divergent motif frequency that we had observed with integrate-and-fire simulations in the stochastic irregular regime, we hypothesize features of this mechanism that may be missing. As mentioned in section 5.1.2, this simple stochastic simulation model does not take refractory period into account. Hence, there is no minimum on the amount of time between active states of a neuron. We are not sure what affect this may have on the mechanism that modulates the influence of divergent motif frequency on event rate, but suspect that it plays a role.

Further, divergent connection motifs provide neurons with common input. When more neurons receive common input, they also tend to receive correlated input, which can lead to increased fluctuations that compound over time, rather than equalize. Thus, we expected to see increased synchrony with increased frequency of divergent connection motifs. This was observed in integrate-and-fire simulations, but not with the simple stochastic simulations. We notice that the simple stochastic model lacks the pulse that occurs when a neuron fires. Instead, when the stochastic model simulates a neuron firing, the neuron remains active for a variable amount of time, meanwhile providing constant input to downstream neurons. This is one indicator that the simple stochastic model does not accurately capture the phenomenon of the integrate-and-fire simulations.

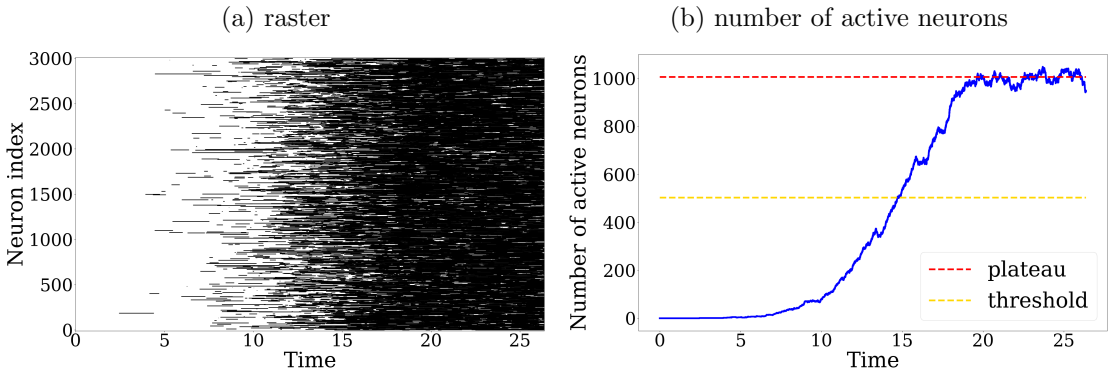


Figure 5.1: Stochastic simulation results. **(a)** A raster plot of a single stochastic simulation on a spatially homogeneous network. Parameters: $\alpha_{\text{recip}} = 0.5$, $\alpha_{\text{conv}} = \alpha_{\text{div}} = \alpha_{\text{chain}} = 0$. **(b)** A plot of the number of active neurons for a single stochastic simulation. With horizontal lines indicating the plateau level and the threshold for an event.

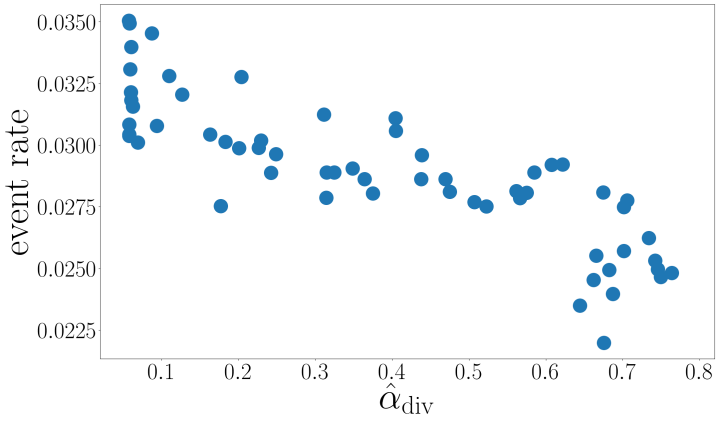


Figure 5.2: A scatter plot of event rate vs. $\hat{\alpha}_{\text{div}}$ from stochastic simulations on sixty spatially homogeneous random networks with α_{div} chosen randomly from $[0, 0.5]$, and all other α values zero ($\alpha_{\text{recip}} = \alpha_{\text{conv}} = \alpha_{\text{chain}} = 0$).

Bibliography

[1] Liqiong Zhao, Bryce Beverlin, Tay Netoff, and Duane Nykamp. Synchronization from second order network connectivity statistics. *Frontiers in Computational Neuroscience*, 5:28, 2011.

[2] Rodrigo Perin, Thomas K. Berger, and Henry Markram. A synaptic organizing principle for cortical neuronal groups. *Proceedings of the National Academy of Sciences*, 108(13):5419–5424, 2011, https://www.pnas.org/content/108/13/5419.full.pdf.

[3] Sen Song, Per Jesper Sjöström, Markus Reigl, Sacha Nelson, and Dmitri B Chklovskii. Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLOS Biology*, 3(3), 03 2005.

[4] Nikolai Axmacher, Florian Mormann, Guillen Fernández, Christian E. Elger, and Juergen Fell. Memory formation by neuronal synchronization. *Brain Research Reviews*, 52(1):170–182, 2006.

[5] Peter J Uhlhaas and Wolf Singer. Abnormal neural oscillations and synchrony in schizophrenia. *Nature reviews. Neuroscience*, 11:100–113, 02 2010.

[6] Hagai Bergman, Ariela Feingold, Asaph Nini, Aeyal Raz, Hamutal Slovin, Moshe Abeles, and Eilon Vaadia. Physiological aspects of information processing in the basal ganglia of normal and parkinsonian primates. *Trends in Neurosciences*, 21(1):32–38, 1998.

[7] Mark A. Kramer, Fen-Lei Chang, Maruice E. Cohen, Donna Hudson, and Andrew J. Szeri. Synchonization measures of the scalp electroencephalogram can

discriminate healthy from alzheimer's subjects. *International Journal of Neural Systems*, 17(02):61–69, 2007, https://doi.org/10.1142/S0129065707000932. PMID: 17565502.

[8] Theoden I. Netoff and Steven J. Schiff. Decreased neuronal synchronization during experimental seizures. *Journal of Neuroscience*, 22(16):7297–7307, 2002, http://www.jneurosci.org/content/22/16/7297.full.pdf.

[9] Wim van Drongelen, Henner Koch, Charles Marcuccilli, Fernando Penã, and Jan-Marino Ramirez. Synchrony levels during evoked seizure-like bursts in mouse neocortical slices. *Journal of Neurophysiology*, 90(3):1571–1580, 2003, https://doi.org/10.1152/jn.00392.2003. PMID: 12750417.

[10] P. Erdős and A. Rényi. On random graphs. i. *Publicationes Mathematicae*, 6:290––297, 1959.

[11] Duane Q. Nykamp, Daniel Friedman, Sammy Shaker, Maxwell Shinn, Michael Vella, Albert Compte, and Alex Roxin. Mean-field equations for neuronal networks with arbitrary degree distributions. *Phys. Rev. E*, 95:042323, Apr 2017.

[12] Chin-Yueh Liu and Duane Q. Nykamp. A kinetic theory approach to capturing interneuronal correlation: the feed-forward case. *Journal of Computational Neuroscience*, 26(3):339–368, Jun 2009.

[13] Robert Rosenbaum, James Trousdale, and Kresimir Josic. Pooling and correlated neural activity. *Frontiers in Computational Neuroscience*, 4:9, 2010.

[14] Robert Rosenbaum, James Trousdale, and Kresimir Josic. The effects of pooling on spike train correlations. *Frontiers in Neuroscience*, 5:58, 2011.

[15] Samantha Fuller. Second order networks with spatial structure. University of Minnesota unpublished master's thesis, June 2016.

[16] Jakob H. Macke, Philipp Berens, Alexander S. Ecker, Andreas S. Tolias, and Matthias Bethge. Generating spike trains with specified correlation coefficients. *Neural Computation*, 21(2):397–423, 2009, https://doi.org/10.1162/neco.2008.02-08-713. PMID: 19196233.

[17] Dan Goodman and Romain Brette. Brian: a simulator for spiking neural networks in python. *Frontiers in Neuroinformatics*, 2:5, 2008.

[18] Y. Kuramoto. *Chemical Oscillations, Waves, and Turbulence*, volume 19 of *Springer Series in Synergetics*. Springer-Verlag Berlin Heidelberg, 1984.

[19] Nicolas Brunel and Vincent Hakim. Fast global oscillations in networks of integrate-and-fire neurons with low firing rates. *Neural Computation*, 11(7):1621–1671, 1999, https://doi.org/10.1162/089976699300016179.

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

Appendix A

Network generation algorithm

Our network generation algorithm is based on the algorithm of Sam Fuller [15]. We translated Fuller's Matlab code to Python and modified it to increase the range of the networks that can be generated. Our most significant changes include altering the definition of Gaussian random variables Z_{ij} and the threshold values θ_{ij} used in the dichotomization process. This allows for a wider range of α values and higher values of p_{ij}, the probability of connection from neuron j to neuron i.

We also increased the computational efficiency of the algorithm from $O(N^3)$ to $O(N^2)$, decreasing runtime and allowing for large networks to be generated quickly. Fuller primarily generated networks of size $N = 300$ to $N = 1000$. Now, with our modifications, we are able to generate networks of size $N = 10,000$ efficiently. For completeness, here is the algorithm that we used:

A.1 Overview

To create networks, we generate the connectivity matrix W. We use a dichotomized Gaussian approach to add minimal structure beyond the desired first and second order statistics (prescribed using p_{ij} and α values as described in Chapter 3) and minimize higher order statistics. To do this, we generate Gaussian random variables Z_{ij}, with specified covariance matrix Σ. Then we threshold each Z_{ij} to create the connectivity

matrix W:

$$W_{ij} = \begin{cases} 1, & \text{if } Z_{ij} > \theta_{ij} \\ 0, & \text{otherwise,} \end{cases}$$

where the threshold value θ_{ij} is chosen such that $P(W_{ij} = 1) = p_{ij}$. Thus, $\theta_{ij} = \infty$ if $p_{ij} = 0$. The covariance matrix Σ is such that we get the desired first and second order statistics of W.

For moderate sized networks, Σ is too large to fit in memory, as it is size $N^2 \times N^2$. So we exploit its special structure to tractably generate the random variables, Z_{ij} with that covariance. The covariance matrix Σ has a simple structure for spatially homogeneous networks, but is destroyed once we allow p_{ij} to depend on location. However, with some assumptions on p_{ij}, and an number of approximations, we generalize the SONET approach to allow for spatially varying p_{ij}.

We follow this process to generate W:

1. Determine the covariance matrix Σ of the Z_{ij} Gaussian random variables. Σ is such that $\text{Var}(Z_{ij}) = 1$, and yields the appropriate second order statistics for W. Thus, Σ depends on p_{ij} and α values.

2. Compute the approximate square root S of Σ. Note that, even for relatively small networks, both Σ and S may be too large for memory, so we exploit their structure and store components in terms of a small number of parameters.

3. Generate a matrix X of standard normal random variables, then let $Z = SX$.

4. Finally, threshold each Z_{ij} to get the connectivity matrix W.

A.2 Calculating the covariance matrix Σ

We determine the entries of the covariance matrix Σ of the Gaussian random variables Z_{ij} such that $\text{Var}(Z_{ij}) = 1$, $\text{Var}(Z_{ii}) = 0$, and yields desired first and second order connectivity statistics as described in 3.1.1:

$$P(W_{ij} = 1) \;=\; E(W_{ij}) \;=\; p_{ij} \tag{A.1}$$

$$\begin{aligned} P(W_{ij}=1,W_{ji}=1) &= p_{ij}p_{ji}(1+\alpha_{\text{recip}}) \\ P(W_{ij}=1,W_{ik}=1) &= p_{ij}p_{ik}(1+\alpha_{\text{conv}}) \\ P(W_{ij}=1,W_{kj}=1) &= p_{ij}p_{kj}(1+\alpha_{\text{div}}) \\ P(W_{ij}=1,W_{jk}=1) &= p_{ij}p_{jk}(1+\alpha_{\text{chain}}). \end{aligned} \tag{A.2}$$

We denote correlation coefficients as follows: $\rho_{ij}^{\text{recip}} = \rho_{ji}^{\text{recip}}$ for the reciprocal motif, $\rho_{ijk}^{\text{conv}} = \rho_{ikj}^{\text{conv}}$ for the convergent motif, $\rho_{ikj}^{\text{div}} = \rho_{kij}^{\text{div}}$ for the divergent motif, and $\rho_{jik}^{\text{chain}}$ for the chain motif. (We refer to a generic correlation coefficient as $\rho_{ij\tilde{i}\tilde{j}}^{\upsilon}$, where $\upsilon \in$ {recip, conv, div, chain}.) Hence, by the assumptions in A.1 and A.2, the nonzero entries of Σ are

$$\begin{aligned} \Sigma_{(i,j),(i,j)} &= \text{Var}(Z_{ij}) \;=\; 1, \text{ for } i \neq j \\ \Sigma_{(i,j),(j,i)} &= \text{Cov}(Z_{ij}, Z_{ji}) \;=\; \rho_{ij}^{\text{recip}} \\ \Sigma_{(i,j),(i,k)} &= \text{Cov}(Z_{ij}, Z_{ik}) \;=\; \rho_{ijk}^{\text{conv}} \\ \Sigma_{(i,j),(k,j)} &= \text{Cov}(Z_{ij}, Z_{kj}) \;=\; \rho_{ikj}^{\text{div}} \\ \Sigma_{(i,j),(j,k)} &= \text{Cov}(Z_{ij}, Z_{jk}) \;=\; \rho_{ijk}^{\text{chain}}. \end{aligned}$$

All remaining entries of Σ are zero.

Now, we can calculate these nonzero entries of Σ using the first and second order network connectivity statistics. Although it is possible to calculate each $\rho_{ij\tilde{i}\tilde{j}}^{\upsilon}$ using a double Gaussian integral, there are $O(N^3)$ $\rho_{ij\tilde{i}\tilde{j}}^{\upsilon}$ values to be that would need to be calculated, as this depends on the neuron label. Fortunately, Fuller discovered an approximation of each $\rho_{ij\tilde{i}\tilde{j}}^{\upsilon}$ and α_{υ} using a nonlinear least-squares fit:

$$\rho_{ij\tilde{i}\tilde{j}}^{\upsilon} \;=\; c_3(p_{ij}+c_1)(p_{\tilde{i}\tilde{j}}+c_1)(\alpha_{\upsilon}+c_2), \tag{A.3}$$

where $c_1 = 0.2658, c_2 = 0.047, c_3 = 1.9531$. This no longer requires us to calculate each $\rho_{ij\tilde{i}\tilde{j}}^{\upsilon}$ individually, as we can now utilize the p_{ij} values. The simplicity of this form allows for further analysis. We define $\widetilde{p}_{ij} = p_{ij} + c_1$, so we can write the nonzero entries of Σ

as

$$\begin{aligned}
\Sigma_{(i,j),(i,j)} &= \mathrm{Var}(Z_{ij}) \;=\; 1 \\
\Sigma_{(i,j),(j,i)} &= \mathrm{Cov}(Z_{ij}, Z_{ji}) \;=\; c_3 \widetilde{p}_{ij} \widetilde{p}_{ji} (\alpha_{\mathrm{recip}} + c_2) \\
\Sigma_{(i,j),(i,k)} &= \mathrm{Cov}(Z_{ij}, Z_{ik}) \;=\; c_3 \widetilde{p}_{ij} \widetilde{p}_{ik} (\alpha_{\mathrm{conv}} + c_2) \\
\Sigma_{(i,j),(k,j)} &= \mathrm{Cov}(Z_{ij}, Z_{kj}) \;=\; c_3 \widetilde{p}_{ik} \widetilde{p}_{jk} (\alpha_{\mathrm{div}} + c_2) \\
\Sigma_{(i,j),(j,k)} &= \mathrm{Cov}(Z_{ij}, Z_{jk}) \;=\; c_3 \widetilde{p}_{ij} \widetilde{p}_{jk} (\alpha_{\mathrm{chain}} + c_2).
\end{aligned}$$

A.3 Equations that determine the square root S

Let S be the square root of the covariance matrix Σ. We assume that S has similar structure as Σ. Thus, we define:

$$\begin{aligned}
S_{(i,j),(i,j)} &= a_{ij} && \\
S_{(i,j),(j,i)} &= b_{ij} \;=\; b_{ji} && \text{(corresponding to reciprocal motif)} \\
S_{(i,j),(i,k)} &= c_{ijk} \;=\; c_{ikj} && \text{(corresponding to convergent motif)} \\
S_{(i,j),(k,j)} &= d_{ijk} \;=\; d_{jik} && \text{(corresponding to divergent motif)} \\
S_{(i,j),(j,k)} &= e_{ijk} && \text{(corresponding to chain motif)}
\end{aligned}$$

with all other entries of S to be zero.

By comparing the entries of S^2 and Σ, we get the following equations:

$$1 = a_{ij}^2 + b_{ij}^2 + \sum_{k \notin \{i,j\}} \left(c_{ijk}^2 + d_{ikj}^2 + e_{ijk}^2 + e_{kij}^2\right) \tag{A.4a}$$

$$\begin{aligned} c_3 \widetilde{p}_{ij} \widetilde{p}_{ji} (\alpha_{\text{recip}} + c_2) &= (a_{ij} + a_{ji}) b_{ij} \\ &\quad + \sum_{k \notin \{i,j\}} (c_{ijk} e_{jik} + c_{jik} e_{ijk} + d_{ikj} e_{kji} + d_{jki} e_{kij}) \end{aligned} \tag{A.4b}$$

$$\begin{aligned} c_3 \widetilde{p}_{ij} \widetilde{p}_{ik} (\alpha_{\text{conv}} + c_2) &= (a_{ij} + a_{ik}) c_{ijk} + b_{ij} e_{jik} + b_{ik} e_{kij} + d_{ikj} e_{ikj} + d_{ijk} e_{ijk} \\ &\quad + \sum_{l \notin \{i,j,k\}} (c_{ijl} c_{ikl} + e_{lij} e_{lik}) \end{aligned} \tag{A.4c}$$

$$\begin{aligned} c_3 \widetilde{p}_{ik} \widetilde{p}_{jk} (\alpha_{\text{div}} + c_2) &= (a_{ik} + a_{jk}) d_{ijk} + b_{ik} e_{jki} + b_{kj} e_{ikj} + c_{ikj} e_{ijk} + c_{jki} e_{jik} \\ &\quad + \sum_{l \notin \{i,j,k\}} (d_{ilk} d_{jlk} + e_{ikl} e_{jkl}) \end{aligned} \tag{A.4d}$$

$$\begin{aligned} c_3 \widetilde{p}_{ij} \widetilde{p}_{jk} (\alpha_{\text{chain}} + c_2) &= (a_{ij} + a_{jk}) e_{ijk} + b_{ij} c_{jik} + b_{jk} d_{ikj} + c_{ijk} d_{ijk} + e_{kij} e_{jki} \\ &\quad + \sum_{l \notin \{i,j,k\}} (c_{jkl} e_{ijl} + d_{ilj} e_{ljk}) \,. \end{aligned} \tag{A.4e}$$

We now have $O(N^3)$ equations with $O(N^3)$ unknowns. We could solve each of these equations, but this would be numerically intractable. Instead we will find an approximate solution to each of these equations.

A.4 Solving for convergent, divergent, and chain entries of S

We will begin by considering equations (A.4c)-(A.4e). For each of these, we assume the first five terms on the right hand side of each equation to be negligible compared to the entire sum on the right hand side, because we are assuming that N is large. By neglecting these terms, each of these equations simplify to be in terms of c_{ijk}, d_{ijk}, and

e_{ijk}, as follows:

$$\begin{aligned} c_3\widetilde{p}_{ij}\widetilde{p}_{ik}(\alpha_{\text{conv}}+c_2) &= \sum_{l\notin\{i,j,k\}}(c_{ijl}c_{ikl}+e_{lij}e_{lik}) \\ c_3\widetilde{p}_{ik}\widetilde{p}_{jk}(\alpha_{\text{div}}+c_2) &= \sum_{l\notin\{i,j,k\}}(d_{ilk}d_{jlk}+e_{ikl}e_{jkl}) \\ c_3\widetilde{p}_{ij}\widetilde{p}_{jk}(\alpha_{\text{chain}}+c_2) &= \sum_{l\notin\{i,j,k\}}(c_{jkl}e_{ijl}+d_{ilj}e_{ljk})\,. \end{aligned}$$

Now, we assume that there are constants c, d, e such that for all i, j, k,

$$c_{ijk} = c\widetilde{p}_{ij}\widetilde{p}_{ik}, \quad d_{ijk} = d\widetilde{p}_{ij}\widetilde{p}_{kj}, \quad e_{ijk} = e\widetilde{p}_{ij}\widetilde{p}_{jk}.$$

So the equations become

$$c_3(\alpha_{\text{conv}}+c_2) = \left(\sum_{l\notin\{i,j,k\}}\widetilde{p}_{il}^2\right)c^2 + \left(\sum_{l\notin\{i,j,k\}}\widetilde{p}_{li}^2\right)e^2 \tag{A.5a}$$

$$c_3(\alpha_{\text{div}}+c_2) = \left(\sum_{l\notin\{i,j,k\}}\widetilde{p}_{lk}^2\right)d^2 + \left(\sum_{l\notin\{i,j,k\}}\widetilde{p}_{kl}^2\right)e^2 \tag{A.5b}$$

$$c_3(\alpha_{\text{chain}}+c_2) = \left(\sum_{l\notin\{i,j,k\}}\widetilde{p}_{jl}^2\right)ce + \left(\sum_{l\notin\{i,j,k\}}\widetilde{p}_{lj}^2\right)de. \tag{A.5c}$$

Now we see that the left hand side of each of these equations does not depend on i, j, k, but the right hand side does. So we let $\widetilde{\Phi}$ be an $N \times N$ matrix such that $\widetilde{\Phi}_{ij} = \widetilde{p}_{ij}^2$, and assume that the row-sums and column-sums of $\widetilde{\Phi}$ are all the same value, which we will call ϕ. Since $\widetilde{p}_{ij} = p_{ij} + c_1$, this assumption is equivalent to $\sum_j (p_{ji}+c_1)^2 = \sum_j (p_{ij}+c_1)^2 = \phi$, for all i. When, p_{ij} is much smaller than c_1, this approximately requires that $\sum_j p_{ij} = \sum_j p_{ji}$ be equivalent for all i. Well, $c_1 = 0.2658$, so we assume that for most desired networks, p_{ij} will be much smaller than c_1. Hence, we can justify the premise that $\sum_j p_{ij} = \sum_j p_{ji}$ is equivalent for all i.

Now, we can see that the coefficients of c, d, e on the right hand side of equations (A.5a)-(A.5c) are approximately ϕ. Thus, we assume that for each of these sums, the two nonzero missing terms are negligible compared to ϕ, because we assume that N is

large. Now, we rewrite the system of equations (A.5) as

$$\begin{aligned} c_3(\alpha_{\text{conv}} + c_2) &= \phi c^2 + \phi e^2 \\ c_3(\alpha_{\text{div}} + c_2) &= \phi d^2 + \phi e^2 \\ c_3(\alpha_{\text{chain}} + c_2) &= \phi ce + \phi de. \end{aligned}$$

Solving this system of equations (for c, d, e), is equivalent to solving

$$MM^T = L \tag{A.6}$$

where

$$M = \begin{bmatrix} c\sqrt{\phi} & e\sqrt{\phi} \\ e\sqrt{\phi} & d\sqrt{\phi} \end{bmatrix} \quad \text{and} \quad L = c_3 \begin{bmatrix} \alpha_{\text{conv}} + c_2 & \alpha_{\text{chain}} + c_2 \\ \alpha_{\text{chain}} + c_2 & \alpha_{\text{div}} + c_2 \end{bmatrix}.$$

Since M and L are both real symmetric matrices, we consider eigenvalue decompostions $L = VDV^{-1}$ and $M = UEU^{-1}$, where both V and U are unitary ($V^{-1} = V^T$ and $U^{-1} = U^T$) and D and E are both diagonal. Now, $MM^T = UE^2U^{-1}$, so satisfying equation (A.6) requires $U = V$ and $E^2 = D$. First, we solve for the diagonal matrix E by letting $E_{ii} = \sqrt{D_{ii}}$. Then, we have that $M = VEV^{-1}$ is a solution to (A.6), and we can solve for the values of c, d, e:

$$c = \frac{M_{11}}{\sqrt{\phi}}, \quad d = \frac{M_{22}}{\sqrt{\phi}}, \quad e = \frac{M_{12}}{\sqrt{\phi}} = \frac{M_{21}}{\sqrt{\phi}}.$$

A.5 Finding the remaining entries of S

Now that we have calculated values for c, d, e, we can return to equations (A.4a) and (A.4b) to solve for a_{ij}, a_{ji} and $b_{ij} = b_{ji}$. So we consider:

$$\begin{aligned} 1 &= a_{ij}^2 + b_{ij}^2 + \widetilde{p}_{ij}^2 \sum_{k \notin \{i,j\}} \left(c^2\widetilde{p}_{ik}^2 + d^2\widetilde{p}_{kj}^2 + e^2\widetilde{p}_{jk}^2 + e^2\widetilde{p}_{ki}^2\right) \\ 1 &= a_{ji}^2 + b_{ij}^2 + \widetilde{p}_{ji}^2 \sum_{k \notin \{i,j\}} \left(c^2\widetilde{p}_{jk}^2 + d^2\widetilde{p}_{ki}^2 + e^2\widetilde{p}_{ik}^2 + e^2\widetilde{p}_{kj}^2\right) \\ c_3\widetilde{p}_{ij}\widetilde{p}_{ji}(\alpha_{\text{recip}} + c_2) &= (a_{ij} + a_{ji})b_{ij} + \widetilde{p}_{ij}\widetilde{p}_{ji} \sum_{k \notin \{i,j\}} \left(ce\widetilde{p}_{ik}^2 + cd\widetilde{p}jk^2 + de\widetilde{p}_{kj}^2 + de\widetilde{p}_{ki}^2\right). \end{aligned}$$

We can now rewrite this equations as

$$a_{ij}^2 \; + \; b_{ij}^2 \; = f_{ij} \tag{A.7a}$$

$$a_{ji}^2 \; + \; b_{ij}^2 \; = g_{ij} \tag{A.7b}$$

$$(a_{ij} + a_{ji})b_{ij} \; = h_{ij} \tag{A.7c}$$

where

$$\begin{aligned}
f_{ij} \; &= 1 - \widetilde{p}_{ij}^2 \sum_{k \notin \{i,j\}} \left(c^2 \widetilde{p}_{ik}^2 + d^2 \widetilde{p}_{kj}^2 + e^2 \widetilde{p}_{jk}^2 + e^2 \widetilde{p}_{ki}^2 \right) \\
g_{ij} \; &= 1 - \widetilde{p}_{ji}^2 \sum_{k \notin \{i,j\}} \left(c^2 \widetilde{p}_{jk}^2 + d^2 \widetilde{p}_{ki}^2 + e^2 \widetilde{p}_{ik}^2 + e^2 \widetilde{p}_{kj}^2 \right) \\
h_{ij} \; &= c_3 \widetilde{p}_{ij} \widetilde{p}_{ji} (\alpha_{\text{recip}} + c_2) - \widetilde{p}_{ij} \widetilde{p}_{ji} \sum_{k \notin \{i,j\}} \left(ce\widetilde{p}_{ik}^2 + cd\widetilde{p}jk^2 + de\widetilde{p}_{kj}^2 + de\widetilde{p}_{ki}^2 \right).
\end{aligned}$$

Note that this requires $f_{ij} > 0$ and $g_{ij} > 0$ for solutions to exist. When solutions exist, we can use equation (A.7c) to rewrite equations (A.7a) and (A.7b) in terms of a_{ij} and a_{ji}:

$$a_{ij}^2 + \frac{h_{ij}^2}{(a_{ij} + a_{ji})^2} \; = f_{ij} \tag{A.8a}$$

$$a_{ji}^2 + \frac{h_{ij}^2}{(a_{ij} + a_{ji})^2} \; = f_{ij}. \tag{A.8b}$$

Next, we can solve equation (A.8a) for a_{ji}:

$$a_{ji} \; = \; \frac{\pm h_{ij}}{\sqrt{f_{ij} - a_{ij}^2}} - a_{ij}. \tag{A.9}$$

Then we can plug this into equation (A.8b) to get an equation in terms of a_{ij}:

$$\begin{aligned}
\left(\frac{\pm h_{ij}}{\sqrt{f_{ij}-a_{ij}^2}} - a_{ij}\right)^2 + f_{ij} - a_{ij}^2 &= g_{ij} \\
\frac{h_{ij}^2}{f_{ij}-a_{ij}^2} - 2a_{ij}\frac{\pm h_{ij}}{\sqrt{f_{ij}-a_{ij}^2}} + f_{ij} &= g_{ij} \\
\frac{h_{ij}^2}{f_{ij}-a_{ij}^2} + f_{ij} - g_{ij} &= 2a_{ij}\frac{\pm h_{ij}}{\sqrt{f_{ij}-a_{ij}^2}} \\
\frac{h_{ij}^4}{(f_{ij}-a_{ij}^2)^2} + 2\frac{h_{ij}^2}{f_{ij}-a_{ij}^2}(f_{ij}-g_{ij}) + (f_{ij}-g_{ij})^2 &= 4a_{ij}^2\frac{h_{ij}^2}{f_{ij}-a_{ij}^2} \\
h_{ij}^4 + 2h_{ij}^2(f_{ij}-a_{ij}^2)(f_{ij}-g_{ij}) + (f_{ij}-a_{ij}^2)^2(f_{ij}-g_{ij})^2 &= 4a_{ij}^2h_{ij}^2(f_{ij}-a_{ij}^2) \\
\Big((f_{ij}-g_{ij})^2 + 4h_{ij}^2\Big)a_{ij}^4 + \Big(h_{ij}^2 + f_{ij}(f_{ij}-g_{ij})\Big)^2 & \\
- \Big(2h_{ij}^2(f_{ij}-g_{ij}) + 2f_{ij}(f_{ij}-g_{ij})^2 + 4h_{ij}^2f_{ij}\Big)a_{ij}^2 &= 0.
\end{aligned}$$

Now, we can use the quadratic formula to calculate a_{ij}^2. If a positive solution exists, then we solve for a_{ij}, also being positive. Next, we calculate a_{ji} using equation (A.9), with sign:

$$\left(\frac{h_{ij}^2}{f_{ij}-a_{ij}^2} + f_{ij} - g_{ij}\right) / \left(2a_{ij}\frac{h_{ij}}{\sqrt{f_{ij}-a_{ij}^2}}\right).$$

Lastly, we solve for b_{ij} using equation (A.7c).

A.6 Generating the corresponding second order network

Now that we have determined all entries of S, the approximate square root of the covariance matrix Σ, we are ready to generate Gaussian random variables then dichotomize to get a connectivity matrix for a network with prescribed first and second order statistics. To do this, we first let X be an $N \times N$ matrix of independent standard normal random variables. Then, we define $Z = SX$, to be a matrix of Gaussian random variables with covariance matrix $S^2 \approx \Sigma$. Given S, which is a sparse matrix, it is easy to calculate the

entries of Z:

$$\begin{aligned} Z_{ij} &= \sum_{m,n=1}^{N} S_{(i,j),(m,n)} X_{mn} \\ &= a_{ij} X_{ij} + b_{ij} X_{ji} \\ &\quad + \sum_{n \notin \{i,j\}} c_{ijn} X_{in} + \sum_{m \notin \{i,j\}} d_{imj} X_{mj} + \sum_{n \notin \{i,j\}} e_{ijn} X_{jn} + \sum_{m \notin \{i,j\}} e_{mij} X_{mi}. \end{aligned}$$

We were able to increase the computational efficiency of Fuller's algorithm by reorganizing how this sum is calculated.

Finally, we dichotomize each Z_{ij} to get a Bernoulli random variable W_{ij}:

$$W_{ij} = \begin{cases} 1 & \text{if } Z_{ij} > \theta_{ij} \\ 0 & \text{otherwise,} \end{cases}$$

where $\theta_{ij} = \sqrt{2}\text{erf}^{-1}(1 - 2p_{ij})$. Note that erf is the Gaussian error function, and we use its inverse. Hence, $\theta_{ij} = \infty$ when $p_{ij} = 0$. The resulting matrix W is the adjacency matrix of the network.

Appendix B

Supplementary Figures

B.1 Simulation Results

Here we provide additional scatter plots of our simulation results. These plots vary based on macrostructure (feed-forward, recurrent, and spatially homogeneous), dynamical regime (regular and irregular), and network motif frequencies.

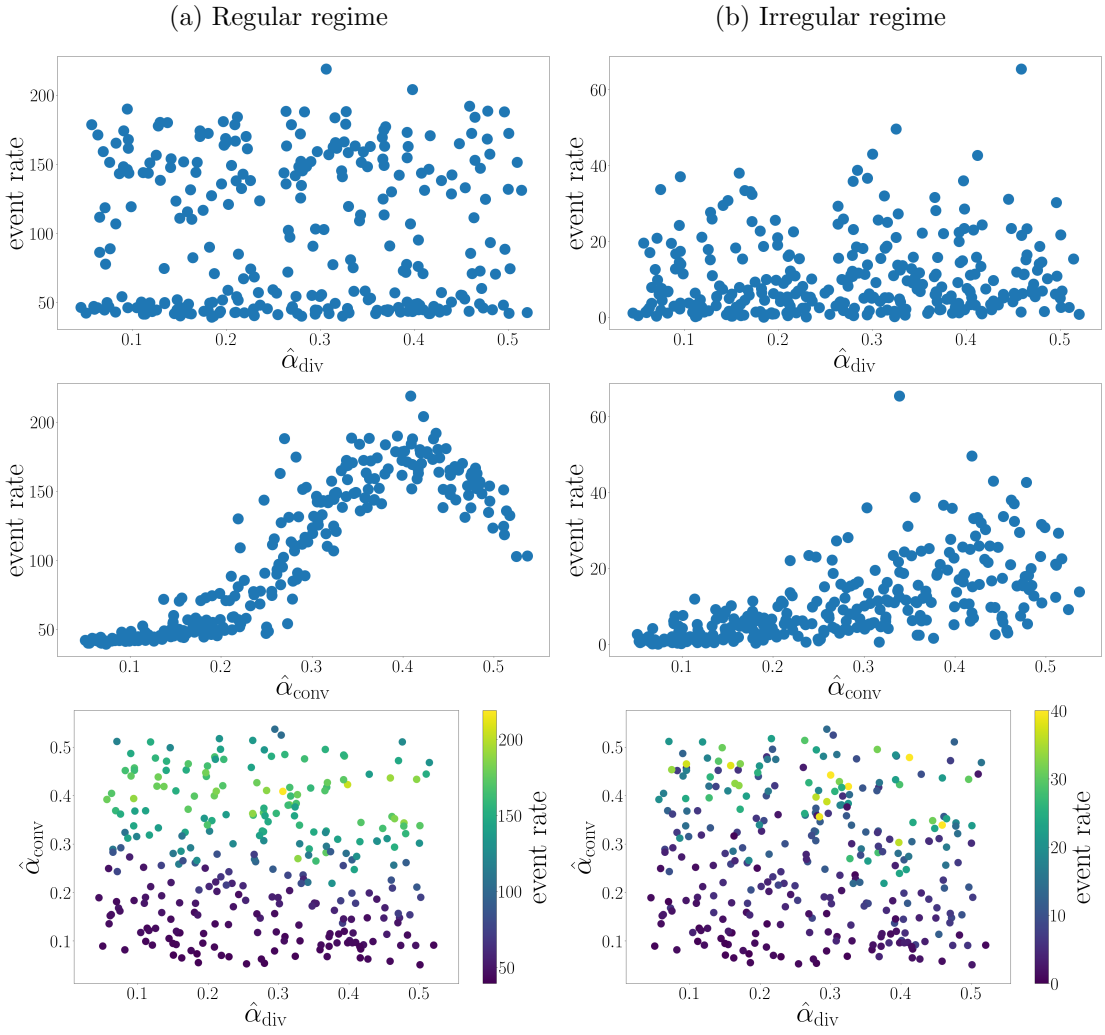


Figure B.1: Scatter plots of results from simulations of 300 recurrent networks with varying convergent and divergent motif frequencies. Each dot represents one simulation. For each network α_{div} and α_{conv} were sampled uniformly from $[0, 0.5]$ and the other α values were zero (i.e., $\alpha_{\text{recip}} = \alpha_{\text{chain}} = 0$). The left column contains plots from simulations done in the regular dynamical regime, and the right plots are from the irregular regime. The top plot is event rate vs. $\hat{\alpha}_{\text{div}}$, the middle plot is event rate vs. $\hat{\alpha}_{\text{conv}}$, and the bottom plot is $\hat{\alpha}_{\text{conv}}$ vs. $\hat{\alpha}_{\text{div}}$ with the color of the dots corresponding to the event rate. Regular regime simulation: Poisson input parameters: $I^{\text{ext}} = 1\text{mV}$, $\lambda = 250$ Hz. Simulation length: 5 seconds. Irregular regime simulation: Poisson input parameters: $I^{\text{ext}} = 1.5\text{mV}$, $\lambda = 113$ Hz. Simulation length: 100 seconds.

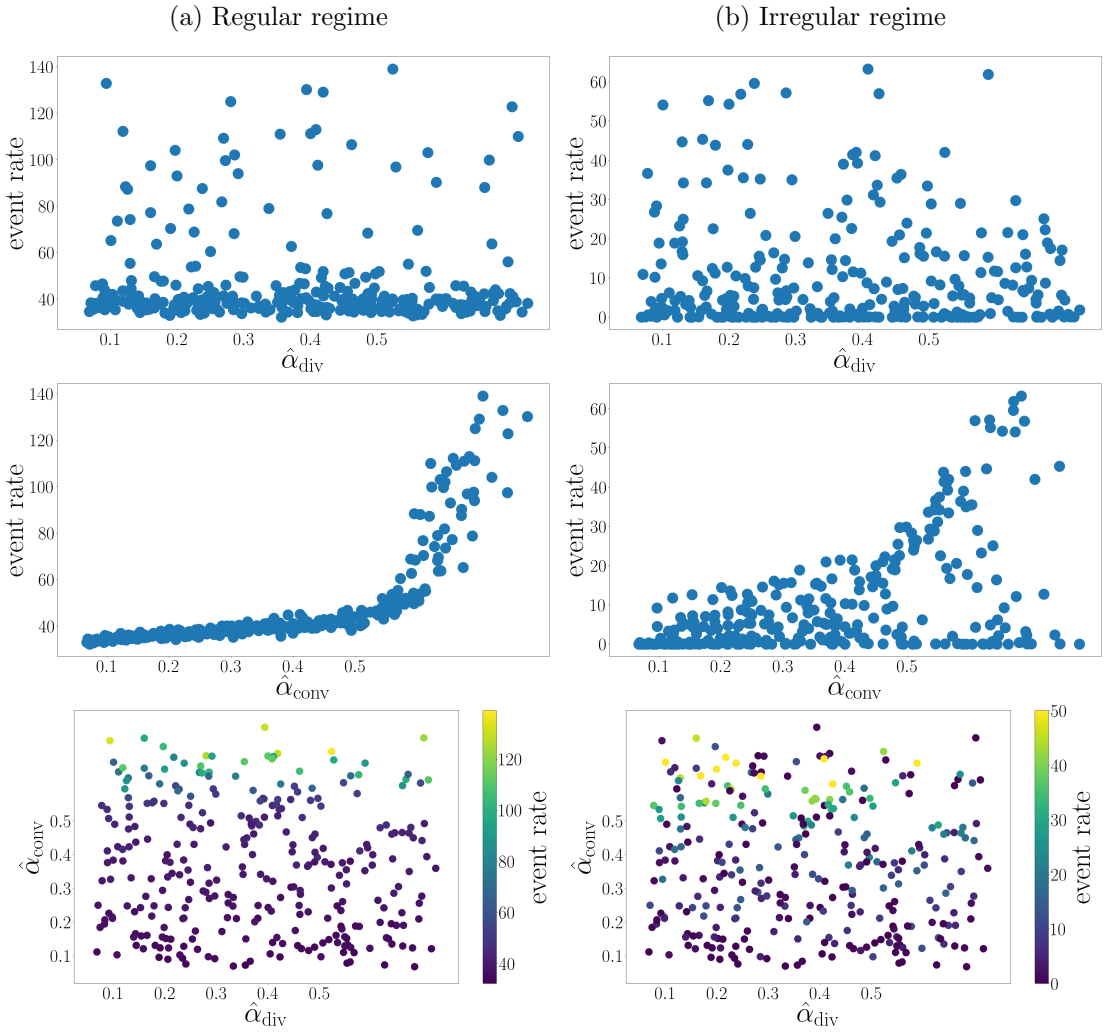


Figure B.2: Scatter plots of results from simulations of 300 spatially homogeneous networks with varying convergent and divergent motif frequencies. Each dot represents one simulation. For each network α_{div} and α_{conv} were sampled uniformly from $[0, 0.5]$ and the other α values were zero (i.e., $\alpha_{\text{recip}} = \alpha_{\text{chain}} = 0$). The left column **(a)** contains plots from simulations done in the regular dynamical regime, and the right plots **(b)** are from the irregular regime. The top plot is event rate vs. $\hat{\alpha}_{\text{div}}$, the middle plot is event rate vs. $\hat{\alpha}_{\text{conv}}$, and the bottom plot is $\hat{\alpha}_{\text{conv}}$ vs. $\hat{\alpha}_{\text{div}}$ with the color of the dots corresponding to the event rate. Regular regime simulation: Poisson input parameters: $I^{\text{ext}} = 1\text{mV}$, $\lambda = 250$ Hz. Simulation length: 5 seconds. Irregular regime simulation: Poisson input parameters: $I^{\text{ext}} = 1.65\text{mV}$, $\lambda = 110$ Hz. Simulation length: 50 seconds.

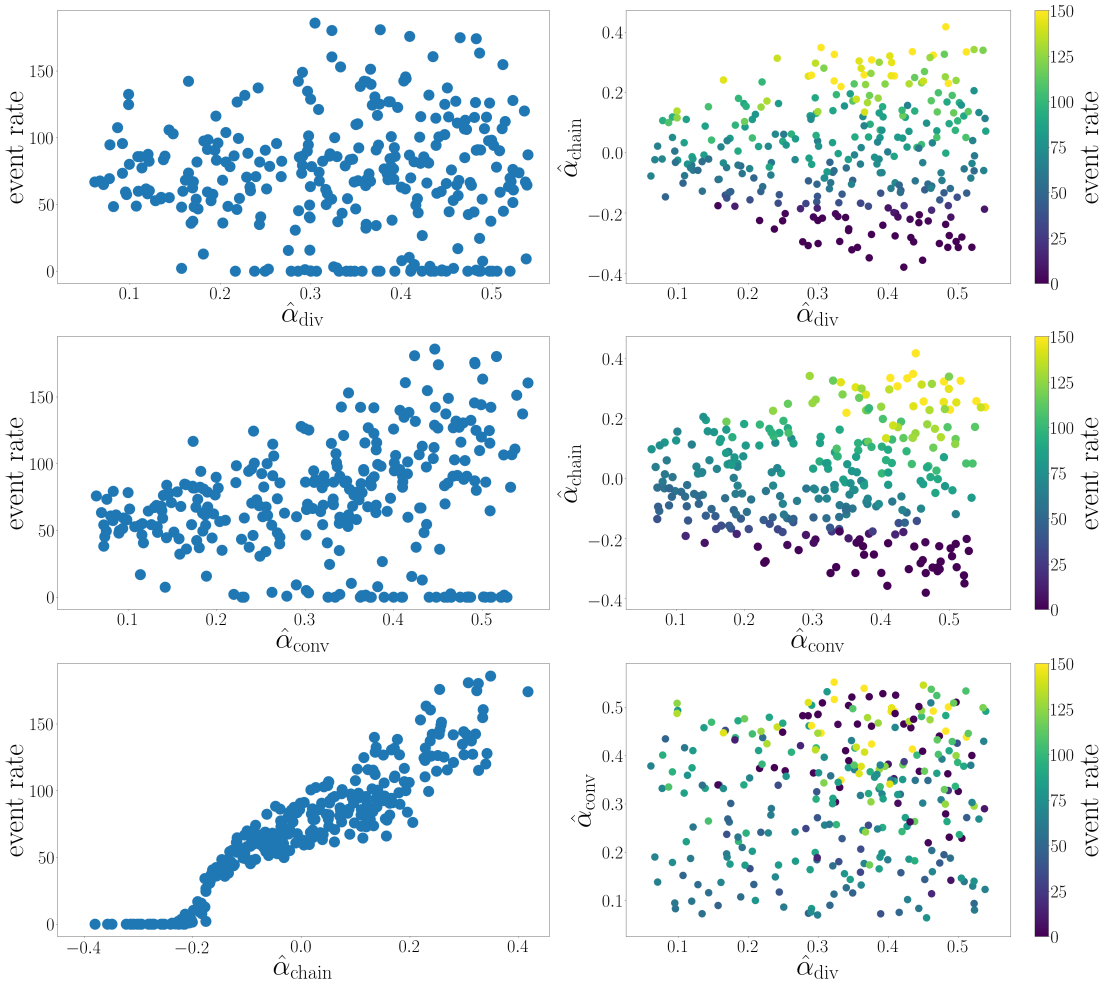


Figure B.3: Scatter plots of results from 300 feed-forward networks with varying convergent, divergent, and chain connections in the regular regime. Each dot represents one simulation. For each network α_{div} and α_{conv} were sampled uniformly from $[0, 0.5]$, α_{chain} was sampled uniformly from $[-0.5, 0.5]$, and α_{recip} was zero. In the left column we have event rate plotted against the three varying α values. The right column plots the different α values against each other with the color of the dots corresponding to the event rate. Poisson input parameters: $I^{\text{ext}} = 1\text{mV}$, $\lambda = 250$ Hz. Simulation length: 5 seconds.

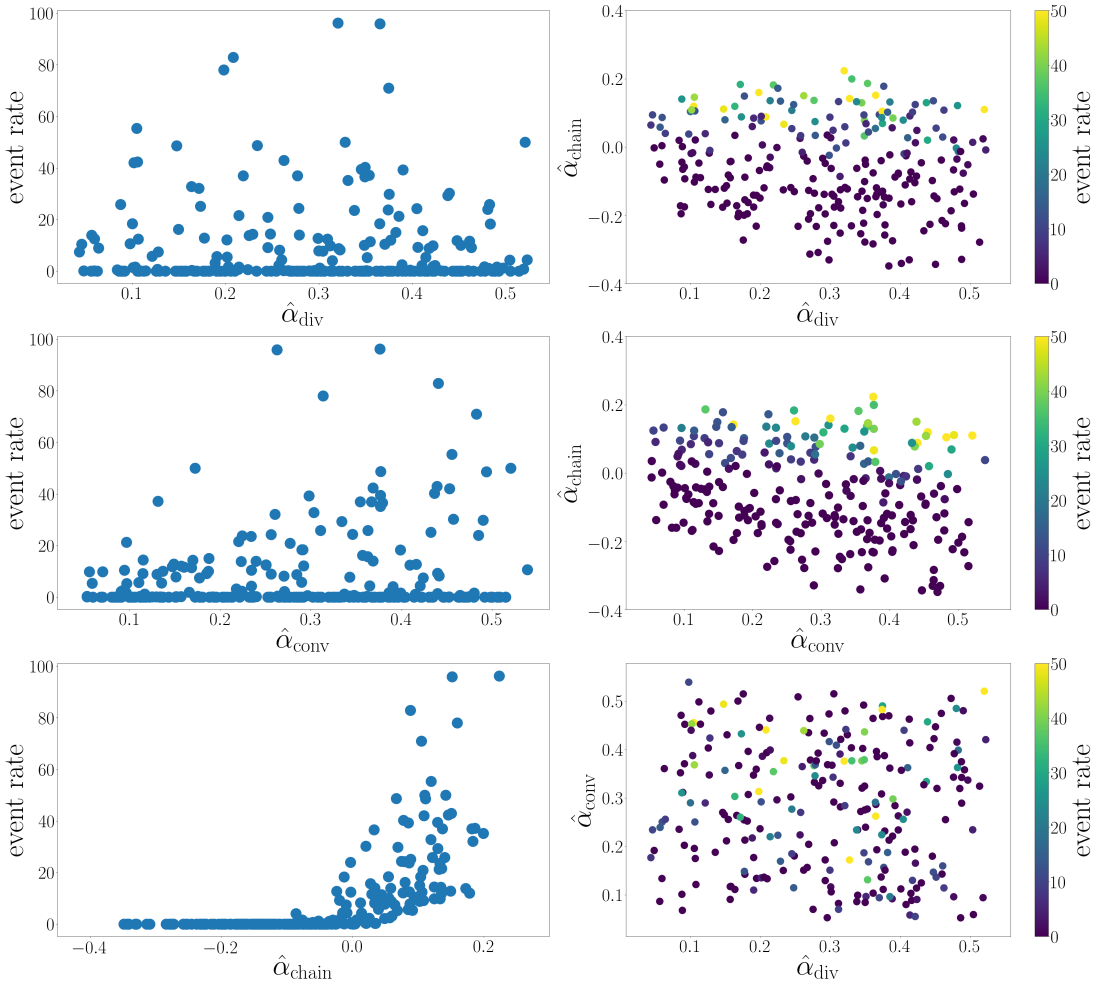


Figure B.4: Scatter plots of results from 300 recurrent networks with varying convergent, divergent, and chain connections in the irregular regime. Each dot represents one simulation. For each network α_{div} and α_{conv} were sampled uniformly from $[0, 0.5]$, α_{chain} was sampled uniformly from $[-0.5, 0.5]$, and α_{recip} was zero. In the left column we have event rate plotted against the three varying α values. The right column plots the different α values against each other with the color of the dots corresponding to the event rate. Poisson input parameters: $I^{\text{ext}} = 1.5\text{mV}$, $\lambda = 113$ Hz. Simulation length: 100 seconds.

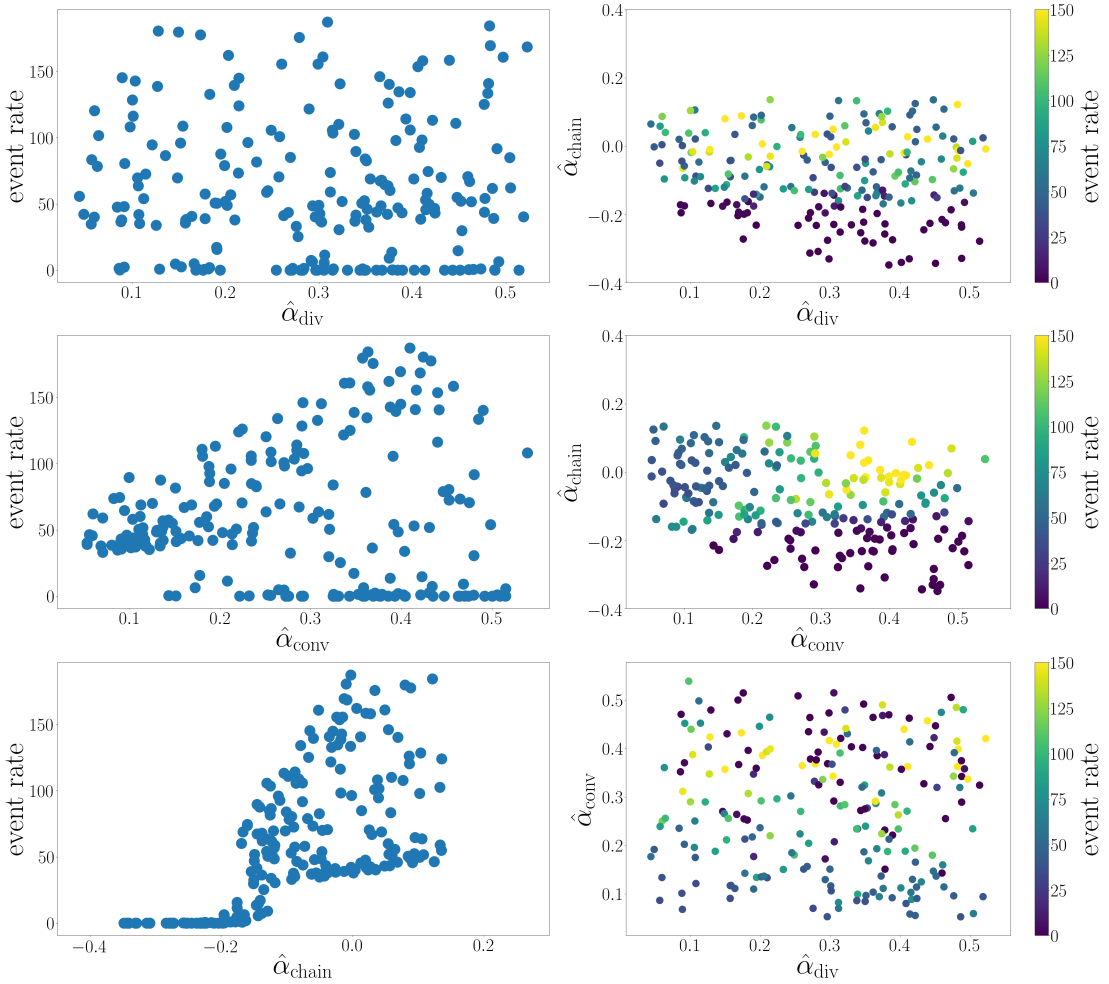


Figure B.5: Scatter plots of results from 300 recurrent networks with varying convergent, divergent, and chain connections in the regular regime. Each dot represents one simulation. For each network α_{div} and α_{conv} were sampled uniformly from $[0, 0.5]$, α_{chain} was sampled uniformly from $[-0.5, 0.5]$, and α_{recip} was zero. In the left column we have event rate plotted against the three varying α values. The right column plots the different α values against each other with the color of the dots corresponding to the event rate. Poisson input parameters: $I^{\text{ext}} = 1\text{mV}$, $\lambda = 250$ Hz. Simulation length: 5 seconds.

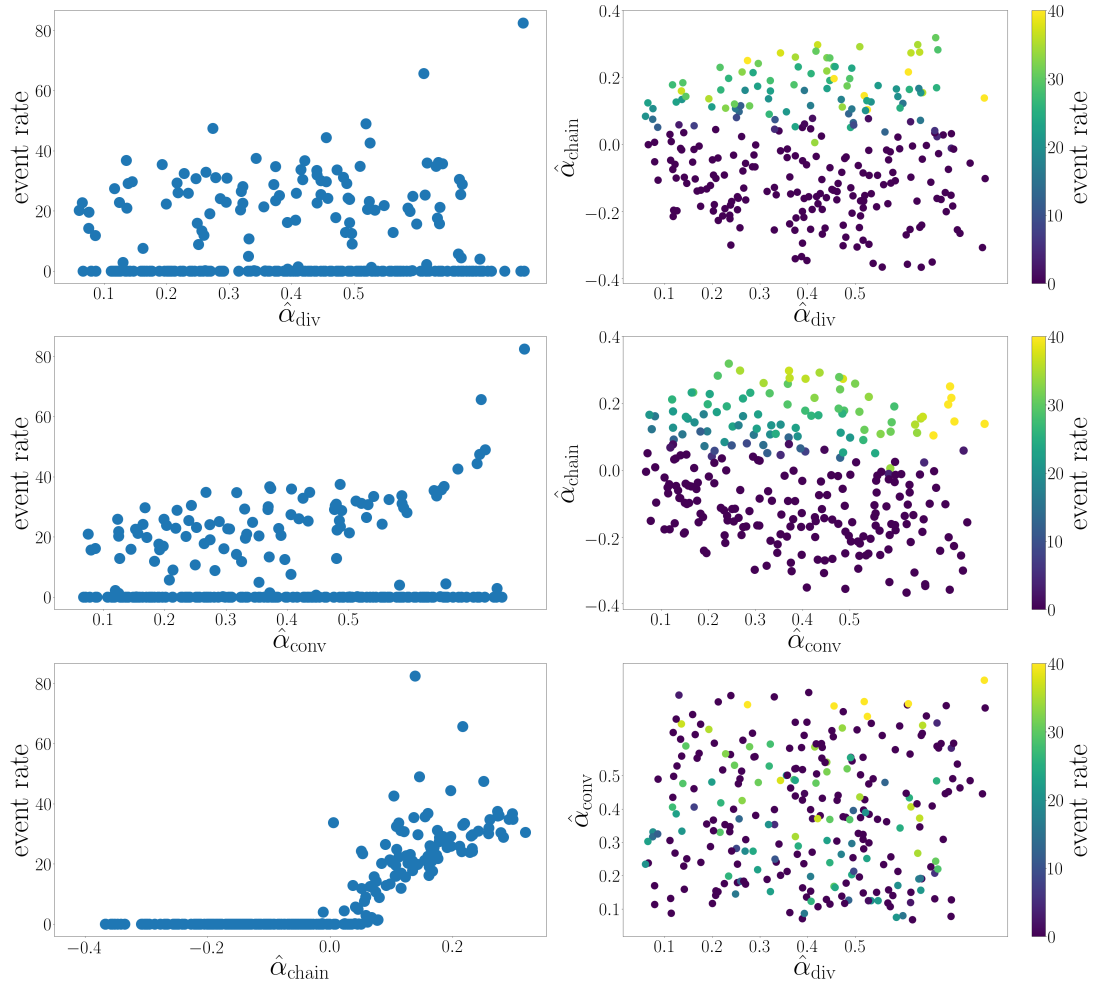


Figure B.6: Scatter plots of results from 300 spatially homogeneous networks with varying convergent, divergent, and chain connections in the irregular regime. Each dot represents one simulation. For each network α_{div} and α_{conv} were sampled uniformly from $[0, 0.5]$, α_{chain} was sampled uniformly from $[-0.5, 0.5]$, and α_{recip} was zero. In the left column we have event rate plotted against the three varying α values. The right column plots the different α values against each other with the color of the dots corresponding to the event rate. Poisson input parameters: $I^{\text{ext}} = 1.65\text{mV}$, $\lambda = 110$ Hz. Simulation length: 50 seconds.

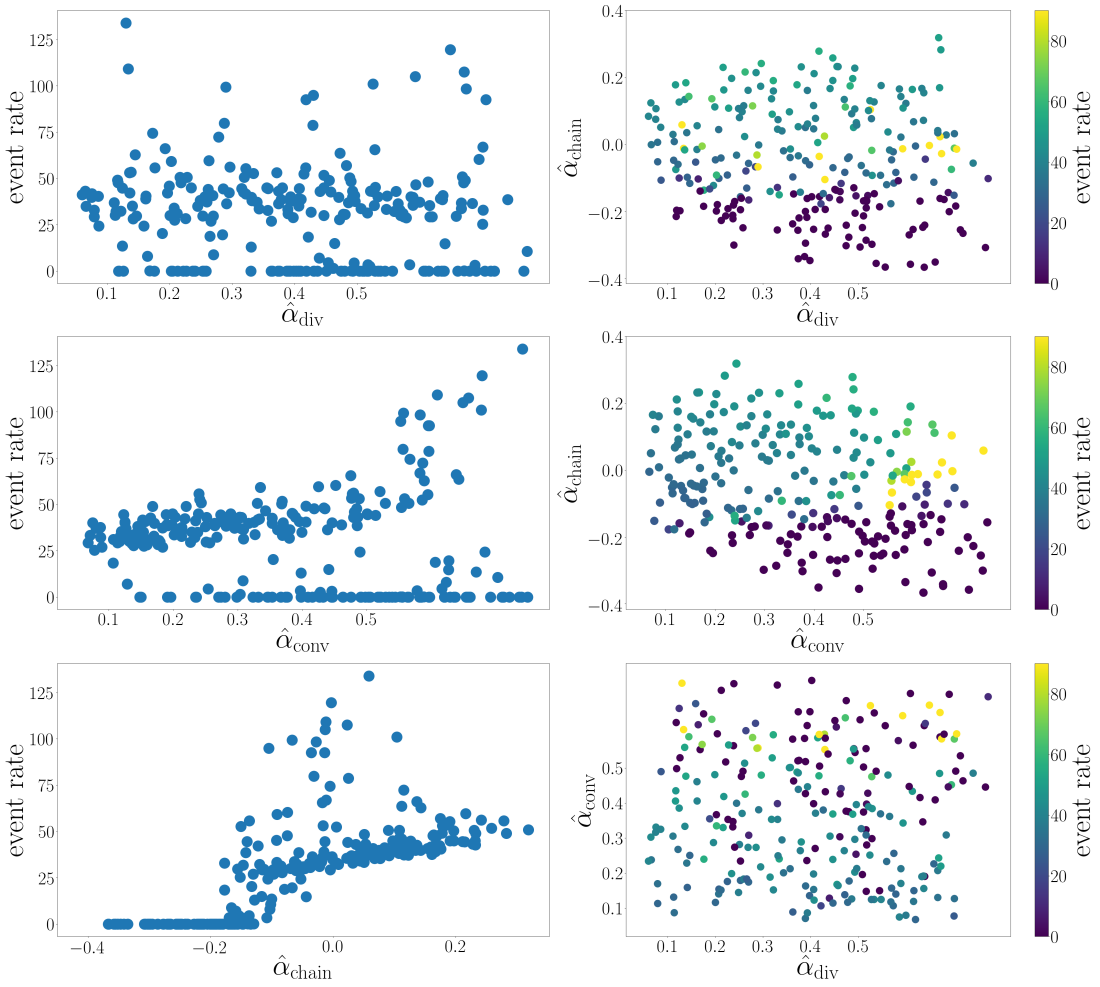


Figure B.7: Scatter plots of results from 300 spatially homogeneous networks with varying convergent, divergent, and chain connections in the regular regime. Each dot represents one simulation. For each network α_{div} and α_{conv} were sampled uniformly from $[0, 0.5]$, α_{chain} was sampled uniformly from $[-0.5, 0.5]$, and α_{recip} was zero. In the left column we have event rate plotted against the three varying α values. The right column plots the different α values against each other with the color of the dots corresponding to the event rate. Poisson input parameters: $I^{\text{ext}} = 1\text{mV}$, $\lambda = 250$ Hz. Simulation length: 5 seconds.