Spatiotemporal Analysis of Neuronal Avalanches in Self-Organized Criticality

Arjun Taneja

A capstone project
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2025

Reading Committee:
Dr. Michael Stiber, Chair
Dr. William Erdly
Dr. Mia Champion

Program Authorized to Offer Degree:
Computing and Software Systems

University of Washington

Abstract

Spatiotemporal Analysis of Neuronal Avalanches in Self-Organized Criticality

Arjun Taneja

Chair of the Supervisory Committee:
Professor and Division Chair Dr. Michael Stiber
Computing and Software Systems

Neural spike activity forms the fundamental basis of information processing and communication in the brain, exhibiting complex spatiotemporal dynamics. Recent advances in computational neuroscience have enabled high-resolution numerical simulations of large-scale neural networks, generating spike-time and location data that capture emergent phenomena such as neuronal avalanches and whole-network bursting. In the case of neuronal avalanches, prior analysis has almost exclusively focused on temporal information, ignoring the spatial information associated with spike data. This project presents an efficient algorithm that incorporates both spatial and temporal constraints for avalanche classification, moving beyond conventional spike-train analyses. Through systematic comparison of temporal-only and spatiotemporal methods, we investigate how the inclusion of spatial information affects avalanche identification and characterization. Finally, we examine how varying spatiotemporal constraints influence the detection and properties of whole-network bursting events, providing insight into the large-scale organization of neural activity.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Professor Michael Stiber, for his patience, unwavering support, and immense wealth of knowledge throughout my capstone project. His guidance and expertise were invaluable in helping me navigate the complexities of this research.

I am grateful to my committee members, Professor Mia Champion and Professor Bill Erdly, for their time, thoughtful feedback, and valuable insights that helped strengthen this work. I would like to acknowledge the Intelligent Networks Laboratory (INL) for providing access to computational resources and data that were essential to this project. The lab meetings and discussions provided a stimulating environment for research development.

Finally, I extend my heartfelt thanks to my family for their constant encouragement and support throughout this journey. Their belief in me has been a great source of strength and motivation.

DEDICATION

To my mother, *Shruti Taneja* and my father, *Rajat Taneja* -
Your unwavering support and belief in me have shaped every step of this journey.
This achievement is as much yours as it is mine.

Chapter 1

BACKGROUND

Neurons in the brain communicate and propagate electrical signals through intricate networks and neural pathways. This electrical activity (referred to as “spikes”) forms the basis of the brain’s communication and computational processes, enabling complex functions such as sensory processing, motor control and higher cognition. Such spiking activity is far from random and is known to exhibit rich spatiotemporal dynamics [6]. Neuron activity at the individual level is a largely binary event i.e. it presents itself in the form of the presence or absence of a spike. The magnitude of spikes is stereotypical and is therefore assumed to carry minimal information. Rather, information is considered to be encoded in the rate or timing of these spikes. The analysis of this spike-time activity offers a deeper look into neural network dynamics.

Spike-time data has been extracted using a variety of experimental and computational methods, including multi-electrode arrays (MEAs) from living cortical cultures [20, 8, 16] and numerical simulations of neuronal networks [23, 1, 19]. With the advancement of technology and computational power, numerical simulations have grown more sophisticated and offer access to higher resolution spike-time and location data at larger scales compared to many experimental approaches.

Among the emergent behaviors observed in these neuronal networks, two prominent features stand out: neuronal avalanches and whole-network bursting.

Neuronal avalanches are described as surges of spike activity that occur more frequently than the system’s average activity. This phenomenon has been observed in both living cortical cultures [4] and numerical simulations [1], and is characterized by a power-law relationship between avalanche size and probability:

$$P(S) \sim S^{\beta} \tag{1.1}$$

Here, $P(S)$ represents the probability distribution of avalanche sizes (S), β represents the power-law exponent/slope of line and $\sim$ denotes proportionality. This power-law behavior is a characteristic trait of systems in a state of self-organized criticality [15, 2, 3]. Such systems are sensitive to small perturbations that can trigger cascades of events through the entire system. Self-organized criticality is a common feature of large, complex natural systems composed of many interacting components such as earthquakes [9] and nuclear chain reactions [10].

Another feature observed as part of the neural dynamics in cultured cortical networks is whole-network bursting. A burst is defined as a synchronized spiking event that involves most or all of the neurons in the network, and are known to play an important role in information processing [14, 11]. Bursts have been shown to propagate outwards from varying origin locations in the form of waves of activity [13], and their timing and intensity are known to change as the culture develops and matures [22, 7].

1.1 Objectives

1. Historically, there has been a focus on analysis and characterization of neuronal avalanches by only taking into account their temporal distributions (referred to as ”spike-train” data). In this project, we present an efficient algorithm that imposes a spatiotemporal constraint, making use of both the spatial and temporal data available in an attempt to offer potentially more accurate avalanche classification.

2. With the spatiotemporal algorithm implemented, we examine how avalanche classification and interpretation differ between the temporal-only and spatiotemporal methods.

3. We also investigate the phenomenon of whole-network bursting to observe how burst analysis results change with the application of different spatiotemporal constraints.

Chapter 2
DATA ACQUISITION

2.1 Data

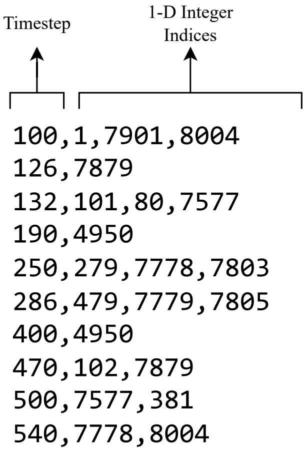
This work utilized a simple computational model of cortical culture development proposed by Kawasaki and Stiber [12], which incorporates the minimal dynamics necessary to reproduce network behaviors observed in living cortical systems through neurite outgrowth and synapse formation.

The model was implemented using Graphitti, a graph-based systems simulator designed to leverage GPU architecture for computationally intensive, large-scale simulations [17].

The simulated network consisted of 10,000 neurons arranged in a structured 100 × 100 rectangular grid, with neurons classified as inhibitory, excitatory, or endogenously active and distributed according to the layout described in [12]. The simulation ran for 600 million time steps at 0.1ms temporal resolution, representing 28 days of *in vitro* development, while also recording the location of each spike to provide spatial resolution.

Kawasaki and Stiber [12] also demonstrated that this network architecture produces stationary bursting behavior under specific parameter combinations, particularly with target firing rates (ϵ) of 1.0Hz or 1.9Hz and excitatory cell fractions of 90% or 98%. For this study, we selected the simulation with $\epsilon = 1.0$Hz firing rate and 90% excitatory neurons as the representative dataset for spatiotemporal analysis.

The spiking data is organized as in Fig. 2.1. The first column in each row indicates the timestep, with each subsequent column containing the 1-dimensional indices of all neurons that generated spikes during that timestep. Our selected simulation encompassed 570,189,562 spikes. For analytical purposes, this study focused exclusively on the final quarter of the simulation data, corresponding to the last seven days of *in vitro* simulated



```
100,1,7901,8004
126,7879
132,101,80,7577
190,4950
250,279,7778,7803
286,479,7779,7805
400,4950
470,102,7879
500,7577,381
540,7778,8004
```

Figure 2.1: Input data format.

development. This period represents the state when the neural network achieved structural maturity and was known to exhibit stationary bursting [12]. This period of the simulation contained 172,323,192 total spikes, and served as the input fed to the avalanche detection algorithms.

2.2 Hardware specifications

Our analysis was run on in-house servers available to Intelligent Networks Laboratory (INL) members. The servers, namely 'Otachi' and 'Raiju', house the simulation data extracted from Graphitti. Our code was run on Otachi, with the hardware specifications detailed in Table 2.1:

Table 2.1: System Hardware Specifications

Category	**Component**	**Specification**
System	Architecture	x86_64
	Memory (RAM)	191 GB total
CPU	CPU Model	Intel Xeon Gold 5118 @ 2.30GHz
	Cores / Threads	24 cores, 48 threads
	Clock Speed	1.0 GHz (Base) – 3.2 GHz (Max)
	Caches (L1/L2/L3)	32K / 1024K / 16.9MB
	Virtualization	VT-x
GPU	GPU Model	NVIDIA Tesla V100-PCIE
	GPU Count	2
	GPU Memory	16 GB per GPU (32 GB total)
	CUDA Version	11.2
	Driver Version	460.32.03

Chapter 3

TEMPORAL AVALANCHE CLASSIFICATION

The temporal avalanche detection algorithm serves as the baseline approach for neuronal avalanche classification. It follows established protocols from existing literature, where avalanches are formed based solely on the temporal spacing between consecutive spike events [13, 4] i.e. spikes are consolidated into avalanches when the inter-spike interval is less than the mean inter-spike interval across the entire network [19, 23].

3.1 Algorithm and Complexity Analysis

The detection process follows a sequential scanning approach through the chronologically ordered spike data. The algorithm maintains a running count of avalanches and compares the temporal distance between consecutive spike events. The inter-spike-interval (ISI) is defined as the time interval between two adjacent spikes in the spike-train. For a spike i represented as (t_i, n_i), where t_i represents its time of occurrence and ($n_i \in [1, 10000]$) represents its 1-dimensional index as governed by its location in the 100×100 rectangular grid of neurons; Δt_i is the ISI between spikes $(i + 1)$ and i:

$$\Delta t_i = t_{i+1} - t_i \tag{3.1}$$

For spike-train data with N total spikes, the mean ISI can be calculated by dividing the total duration of the simulation, by the number of inter-spike intervals:

$$\mathit{meanISI} = \frac{t_N - t_0}{N - 1} \tag{3.2}$$

For our data, the *meanISI* was calculated to be 1.5 timesteps.[1] A gap larger than

[1] Although the *meanISI* is calculated to be 1.5 timesteps ($0.15ms$), the input data contains integral

the *meanISI* between two consecutive spikes represents the end of the previous running avalanche, and the beginning of a new potential avalanche. All consecutive spikes occurring within the *meanISI* window are assigned to the current avalanche regardless of their spatial distribution across the neural network (see Algorithm 1). After all spikes have been consolidated into avalanches, we discard any avalanches of *size* = 1 as they're considered to be isolated single-spike activity i.e. noise.

Algorithm 1 Temporal-only avalanche classification algorithm

```
Input: Spike data $S = \{(t_1, n_1), (t_2, n_2), \ldots, (t_m, n_m)\}$ sorted chronologically by timestep $t_i$, temporal threshold (meanISI) $\tau$
Output: Set of avalanches $A = \{A_1, A_2, \ldots, A_k\}$ where each $A_i$ contains spikes belonging to avalanche $i$
$avalanche\_id \leftarrow 0$
$prev\_timestep \leftarrow -\infty$
$avalanches \leftarrow \emptyset$
for each timestep $t$ in chronological order do
    if $prev\_timestep + \tau < t$ then
        $avalanche\_id \leftarrow avalanche\_id + 1$
    end if
    for each spike $(t, n)$ occurring at timestep $t$ do
        $avalanches[avalanche\_id] \leftarrow avalanches[avalanche\_id] \cup \{(t, n)\}$
    end for
    $prev\_timestep \leftarrow t$
end for
for each avalanche $A_i \in avalanches$ do                ▷ Discard 1-sized avalanches
    if $|A_i| = 1$ then
        $avalanches \leftarrow avalanches \setminus \{A_i\}$
    end if
end for
return $avalanches$
```

The best-case time complexity of the algorithm is $\Omega(N)$, where N is the total number of spikes in the dataset. This occurs when each spike falls outside the temporal threshold τ of its neighbors, causing every spike to initiate a new avalanche. In this scenario, each spike is inserted into a singleton set, which is implemented as an ordered data structure

timesteps. Given that, the same avalanche classification results were categorically achieved by setting the *meanISI* to 1.0 in the avalanche classification code.

in C++ (*std::set*). Each insertion into a small or singleton set has an amortized cost of $O(\log 1) = O(1)$, leading to linear overall processing.

The worst-case time complexity is $O(N \log N)$, which occurs when all spikes fall within the temporal window and form a single massive avalanche. Since each spike is inserted into the same ordered set, the insertion cost grows as the set increases in size: the i^{th} spike costs $O(\log i)$ to insert. Summing over all N spikes, this leads to a total cost of:

$$\sum_{i=1}^{N} \log i \approx O(N \log N) \tag{3.3}$$

This is because the sum $(\log 1 + \log 2 + \ldots + \log N)$ is $\log N!$, which by Stirling's approximation is $N \log N$ [18].

Additionally, the final pass over the avalanche sets to remove singleton events contributes a negligible $O(K)$ time cost, where K is the number of 1-sized avalanches (bounded above by N), and does not change the overall complexity classification.

Space complexity remains $O(N)$ in all cases, as each spike is stored exactly once. Thus, the algorithm is highly efficient for typical neural data with well-separated avalanche events but gracefully handles extreme cases as well.

3.2 Power-law relationship

Fig. 3.1 shows the avalanche size distribution as classified by the temporal-only algorithm for the last 1/4th duration of the simulation data (corresponding to the last 7 of 28 days *in vitro*). We observe a clear distinction between the avalanches with greater than 10^4 spikes, representing whole-network bursts, and the non-burst avalanches with less than 10^3 spikes. The figure also shows a line of best fit to observe the power-law relationship for non-burst avalanches, displaying an exponent of $\beta = -2.52$. A coefficient of determination (describing quality of fit) was also generated, the values for which lie between $R^2 = [0, 1]$, with a higher value indicating a better fit. This value was observed to be 0.937.

After consolidating the spike-train data into avalanches based on their ISIs, the algorithm

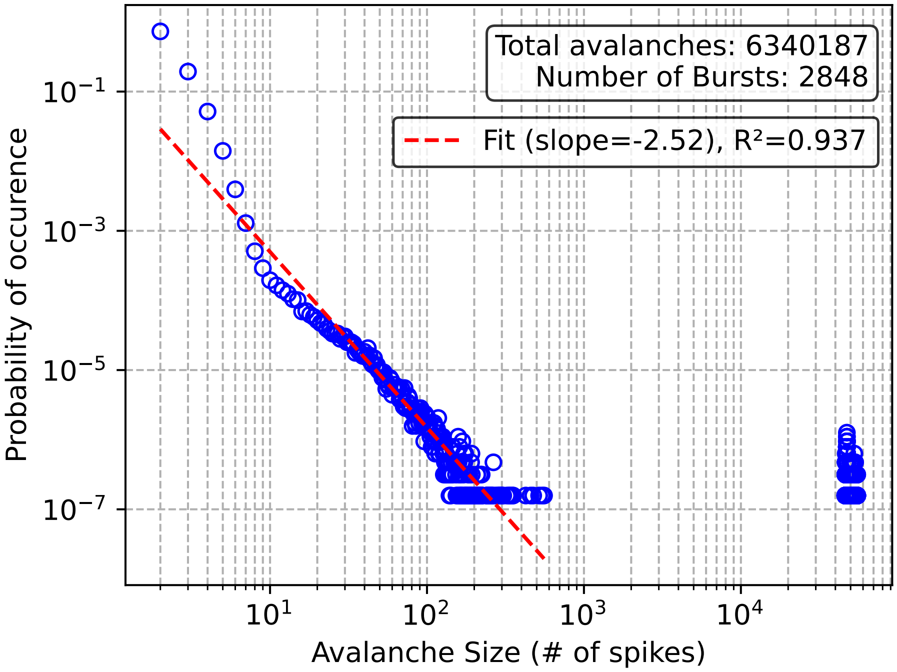


Figure 3.1: Log-log plot of avalanche size-probability with $\tau = 1.5$. We identify avalanches with $> 10^4$ spikes as bursts, and observe a roughly linear size distribution of the non-burst avalanches (size $< 10^3$ spikes) with the slope of line of best fit (β) = -2.52, and coefficient of determination (quality of fit, R^2) = 0.937.

found a total of 6,340,187 temporal avalanches, out of which 2,848 were detected as large-scale bursts. Avalanche detection fundamentally aims to group together causally related spikes. Since this algorithm relies solely on temporal properties and completely disregards spatial information, its bound to group spatially distant neuronal activity—spiking around the same time—into the same avalanche. Given that neural signal propagation requires finite time and occurs through sequential synaptic connections, spikes occurring simultaneously at distant spatial locations cannot be causally related. This temporal-only approach therefore fails to capture the true nature of avalanche distributions.

Chapter 4

SPATIOTEMPORAL AVALANCHE CLASSIFICATION

The spatiotemporal avalanche classification algorithm (see Algorithm 2) is presented as a methodological advancement over the temporal-only approach. Rather than clustering temporally coincident spikes regardless of location, this method incorporates both spatial and temporal proximity constraints to ensure that only causally related spikes are grouped into the same avalanche. By excluding spikes from distant spatial locations that cannot be causally connected, this approach provides potentially more accurate classification of neuronal avalanche events.

However, this clustering approach can lead to a situation where a spike finds itself in spatiotemporal vicinity of two or more distinct avalanche events. For example, we could find a case where two avalanches started in distinct spatial locations, but propagated towards each other, creating an ambiguous situation pertaining to spike ownership.

Lee *et al.*'s approach addresses this by applying spatiotemporal constraints as a post-processing step to the temporal avalanches identified earlier, effectively removing spatially distant outliers while preserving the original avalanche boundaries [13].

In contrast, our approach addresses the problem of ambiguous spike-ownership by merging all avalanches that exhibit spatiotemporal proximity. Rather than maintaining temporal avalanches and discarding spatial outliers, we treat converging avalanches as single unified events.

4.1 Algorithm and Complexity Analysis

For each spike i represented as (t_i, n_i), where t_i represents its time of occurrence and ($n_i \in [1, 10000]$) represents its 1-dimensional index in the 100×100 rectangular grid of neurons, we first extract the corresponding neuron's (x, y) coordinates:

$$x = \left\lfloor \frac{(n_i - 1)}{100} \right\rfloor + 1 \tag{4.1}$$

$$y = n_i - (100 * x) + 100 \tag{4.2}$$

For two spikes i and j, given their 2-D coordinates (x_i, y_i) and (x_j, y_j), we determine their spatial proximity by calculating the Euclidean distance between the neurons associated with the two spikes:

$$\mathit{distance} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{4.3}$$

Given the distances between spiking neurons, the following operations are carried out while parsing each spike in chronological order and allocating it to an avalanche:

1. Sliding Window Maintenance

    - The double-ended queue (*recentSpikes*) holds previous spikes that are in temporal vicinity to the current spike. Spikes that are no longer in temporal proximity are removed from the front of the deque.
    - The time cost of this operation is amortized $O(1)$ per spike, since each spike is added and removed once from the deque and there are a finite number of total spikes, N.

2. Neighbor Scan

    - We find spatiotemporal neighbors by calculating the distance between the current spike i and each of the d_i spikes in the deque $(0 \leq d_i \leq N)$. The time cost of this operation is $O(d_i)$, since the distance is calculated in constant time.

3. Avalanche Insertion and Merging

Algorithm 2 Spatiotemporal Avalanche Detection

```
Input: Spike dataset $S = \{(t_1, n_1), (t_2, n_2), \ldots, (t_m, n_m)\}$ ordered chronologically, temporal threshold $\tau$, spatial radius $r$
Output: Set of avalanches $A = \{A_1, A_2, \ldots, A_k\}$ where each spike in each $A_i$ contains at least one spatiotemporal neighbor in $A_i$
$avalanche\_id \leftarrow 0$
$recent\_spikes \leftarrow \emptyset$                                  ▷ Sliding temporal window (deque)
$spike\_to\_avalanche \leftarrow \emptyset$                            ▷ Spike-avalanche mapping
$avalanches \leftarrow \emptyset$
for each spike $(t, n)$ in chronological order do
    Remove spikes from $recent\_spikes$ where $timestep < t - \tau$
    $neighbors \leftarrow \{s \in recent\_spikes : distance(s.n, n) < r\}$
    if $neighbors = \emptyset$ then
        Add $(t, n)$ to $recent\_spikes$ and continue
    end if
    $overlapping\_avalanches \leftarrow \{spike\_to\_avalanche[s] : s \in neighbors, s \in spike\_to\_avalanche\}$
    if $overlapping\_avalanches = \emptyset$ then
        Create new avalanche with $avalanche\_id$
        Add $(t, n)$ and all neighbors to $avalanches[avalanche\_id]$
        for each spike $s \in neighbors$ do
            $spike\_to\_avalanche[s] \leftarrow avalanche\_id$
        end for
        $spike\_to\_avalanche[(t, n)] \leftarrow avalanche\_id$
        $avalanche\_id \leftarrow avalanche\_id + 1$
    else
        $main\_id \leftarrow$ first element of $overlapping\_avalanches$
        for each $id \in overlapping\_avalanches \setminus \{main\_id\}$ do
            $avalanches[main\_id] \leftarrow avalanches[main\_id] \cup avalanches[id]$
            for each spike $s$ in $avalanches[id]$ do
                $spike\_to\_avalanche[s] \leftarrow main\_id$
            end for
            Remove $avalanches[id]$
        end for
        Add $(t, n)$ and neighbors to $avalanches[main\_id]$
    end if
    Add $(t, n)$ to $recent\_spikes$
end for
return $avalanches$
```

- If none of the spatiotemporal neighbors are part of any avalanches, we create a new avalanche and insert the current spike and its neighbors into it.
- If the spatiotemporal neighbors belong to one unique avalanche, we insert the current spike and all its neighbors to that avalanche.
- If the spatiotemporal neighbors belong to more than one avalanche:
  - We pick one avalanche as the destination.
  - All m spikes from the overlapping avalanches are merged into the destination avalanche, where $0 \leq m \leq N$.
  - The current spike and all its neighbors are inserted into the destination avalanche.
- Since each avalanche is a collection of spikes housed in a C++ *std::set*, each insertion costs $O(\log s)$, where s ($0 \leq s \leq N$) is the size of the set at the time of insertion. The process of merging avalanches involves inserting m spikes from overlapping avalanches into one common avalanche, and the time cost for this merge is given by $O(m \log s)$.

Therefore, the total time cost associated with parsing a spike is given as:

$$T_i = O(d_i + m \log s) \tag{4.4}$$

In the best case, no spike is within spatiotemporal proximity of any other spike. Therefore, for each spike i: $d_i = 0$ since there are no neighbors to be found. Since there are no neighbors, no avalanches are ever formed i.e. $m = s = 0$. Thus, $T_i = O(1)$ per spike. Over N total spikes, the best-case time complexity is:

$$\sum_{i=1}^{N} T_i = \Omega(N) \tag{4.5}$$

In the worst case, processing each spike i triggers repeated merging of multiple avalanches. Since spikes are processed chronologically, and each spike only looks backwards in time to find spatiotemporal neighbors, spike i could have a maximum of $i-1$ neighbors.

The cost of scanning for neighbors in the worst-case is:

$$\sum_{i=1}^{N} d_i = \sum_{i=1}^{N}(i-1) = \frac{N(N-1)}{2} = O(N^2) \tag{4.6}$$

If each spike causes all previous spikes to merge into a new avalanche (in the worst-case), the total merge cost is:

$$\sum_{i=1}^{N}(i-1)\log(i-1) \approx O(N^2 \log N) \tag{4.7}$$

Thus the total worst-case time complexity is $O(N^2 + N^2 \log N) \approx O(N^2 \log N)$.

The space complexity remains $O(N)$ in all cases. Each spike is stored once and assigned to exactly one avalanche set. Thus, all auxiliary data structures grow linearly with N and do not impact asymptotic space bounds.

4.2 Spatiotemporal Parameters

4.2.1 Tau (τ) = 1.5, Radius (r) = 8

The spatial constraint is derived from the Graphitti simulation's connection radius of 2 units, with the constraint set to four times this value to capture neurons where their spike-activity directly influences other neurons in the vicinity i.e. $r = 8$. With the addition of a spatial constraint, the number of total avalanches decreases to 473,509. The dramatic decrease in avalanche count provides evidence that many temporal-only avalanches contained spatially dispersed spikes that were incorrectly grouped together. We still observe the same number of bursts i.e. 2,848. Fig. 4.1 shows that the avalanche size distribution maintained a power-law relationship with an exponent of -2.33, with an R^2 score of 0.953. The improved quality of fit compared to the temporal-only approach provides evidence that spatiotemporal means of clustering avalanches offer results that are closely consistent with a power-law relationship, as predicted by general avalanche theory.

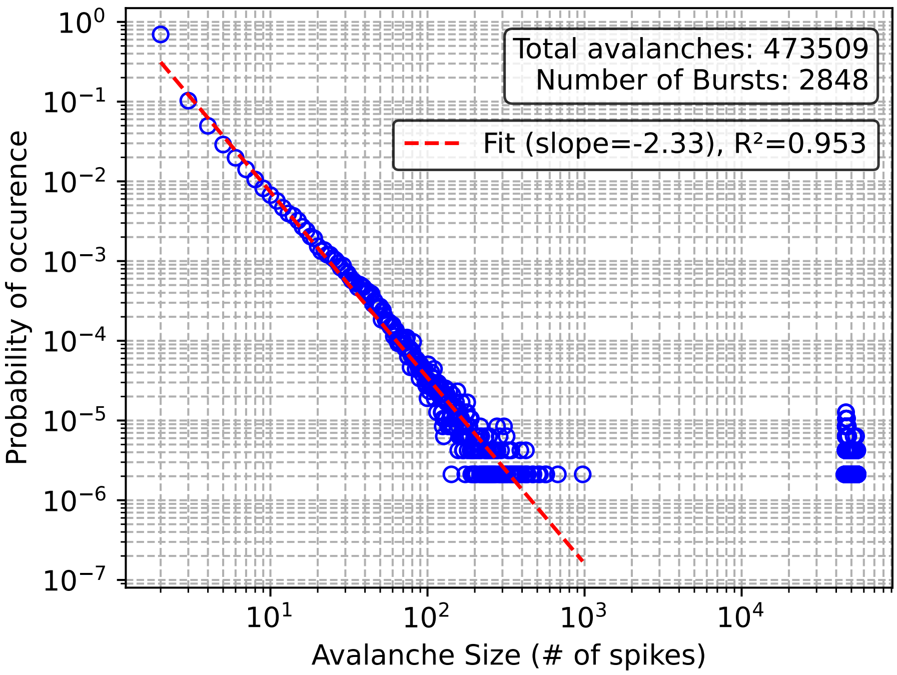


Figure 4.1: Spatiotemporal avalanche size probability distribution with $\tau = 1.5$, $r = 8$. The spatiotemporal algorithm identifies burst and non-burst avalanches, with the non-burst avalanches displaying a power-law exponent = -2.33 and quality of fit $R^2 = 0.953$.

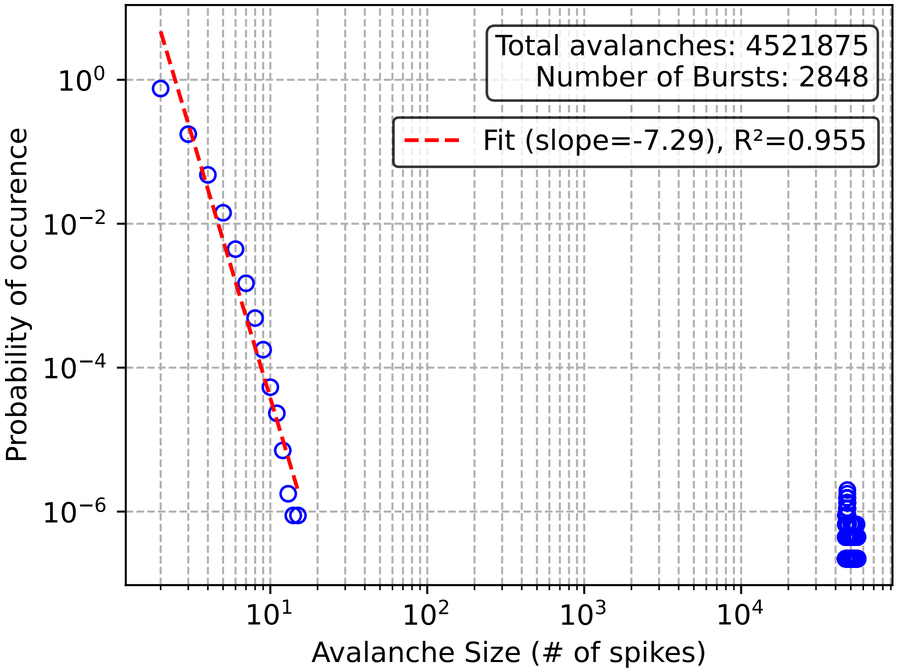


Figure 4.2: Spatiotemporal avalanche size probability distribution with $\tau = 50$, $r = 8$ and slope = -7.29. With adjusted spatiotemporal constraints, we observe the complete disappearance of avalanches sized 16–1000, and also observe a $\approx$ 10x increase in total avalanches found compared to the $\tau_{1.5}$ spatiotemporal analysis.

4.2.2 Tau (τ) = 50, Radius (r) = 8

Instead of using the meanISI calculated from the entire simulation to define temporal thresholds (as done above), a more principled approach would build avalanches using spikes within a spatiotemporal neighborhood. Specifically, the temporal window should reflect the mean ISI for neurons that fall within the spatial window itself. For example, consider a circular spatial window with radius 8: this encompasses approximately $\pi r^2 \approx 200$ neurons. Given that individual neurons spike at an average rate of 1 Hz in our simulation, this group of 200 neurons collectively produces 200 spikes/s. The expected inter-spike interval within this local population is therefore $1/200s \rightarrow 5ms$ (or 50 timesteps).

With these constraints, the algorithm identified 4,521,875 avalanche events while maintaining the same 2,848 burst occurrences. The avalanche distribution is shown in Fig. 4.2, where we observe the complete disappearance of avalanches with size $16 \leq avalSize \leq 1000$ (henceforth referred to as mid-sized avalanches), and observe a much deeper slope of -7.29, indicating a dramatic shift in the avalanche distribution profile.

This result was unexpected, because moving from a smaller temporal constraint to a larger temporal constraint while keeping the spatial constraint constant, should group more spikes together into each avalanche. We're essentially casting a bigger net over the data and since we have a finite number of spikes, we should expect more spikes per avalanche and less total avalanches due to increased merging, but our results show the opposite. We provide an explanation for these results below:

The substantial increase in total avalanches (from 473,509 to 4,521,875) can be explained by the broader temporal grouping enabled by the larger τ_{50} window. With the restrictive $\tau_{1.5}$ window, many spikes—having temporal intervals slightly larger than 1.5 timesteps—remain isolated and don't meet the avalanche detection criteria. The τ_{50} window captures these temporally dispersed spikes and groups them into avalanches. This expanded temporal net transforms previously isolated single spikes and small spike clusters into detectable avalanche events, dramatically increasing the total count.

However, to explain the disappearance of all the mid-sized avalanches, we need to take a closer look at the burst phenomenon.

Chapter 5

BURST ANALYSIS

Bursts are different from avalanches in a couple key ways:

- Bursts contain greater than 10^4 spikes, compared to standard avalanches that cap out at $\approx 10^3$ spikes.

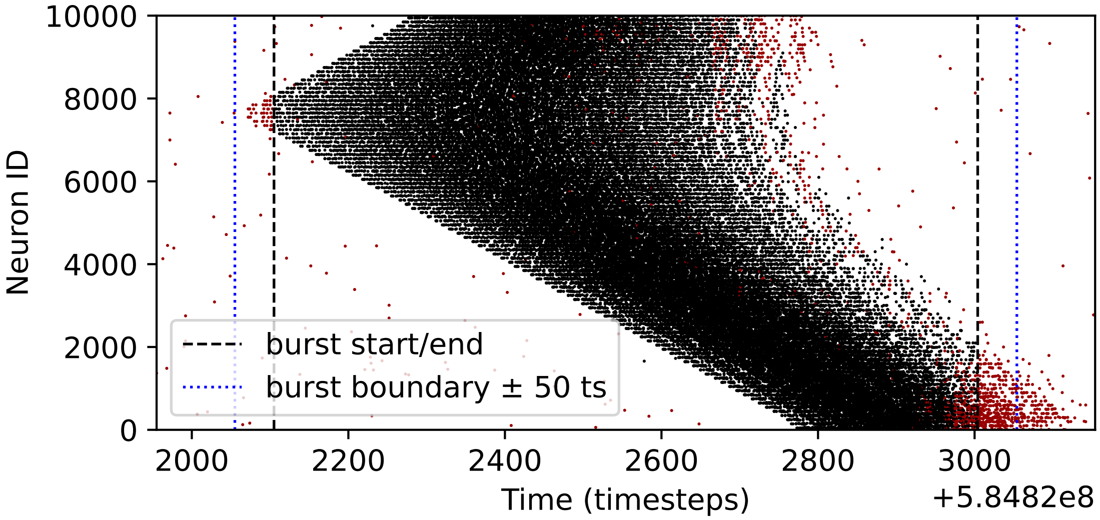
- Bursts manifest in the form of spatially contiguous waves that propagate outwards from varying origin points as the simulation matures [13].

Even though both spatiotemporal analyses identified the same number of bursts (2,848), there are subtle differences in their size and duration.[1] The application of a larger temporal window, τ_{50}, resulted in an increase in both burst size and duration compared to the more restrictive window, $\tau_{1.5}$. The average burst size increased from 48,488 to 49,335 spikes, while the average burst duration grew from 1,109 to 1,346 timesteps.

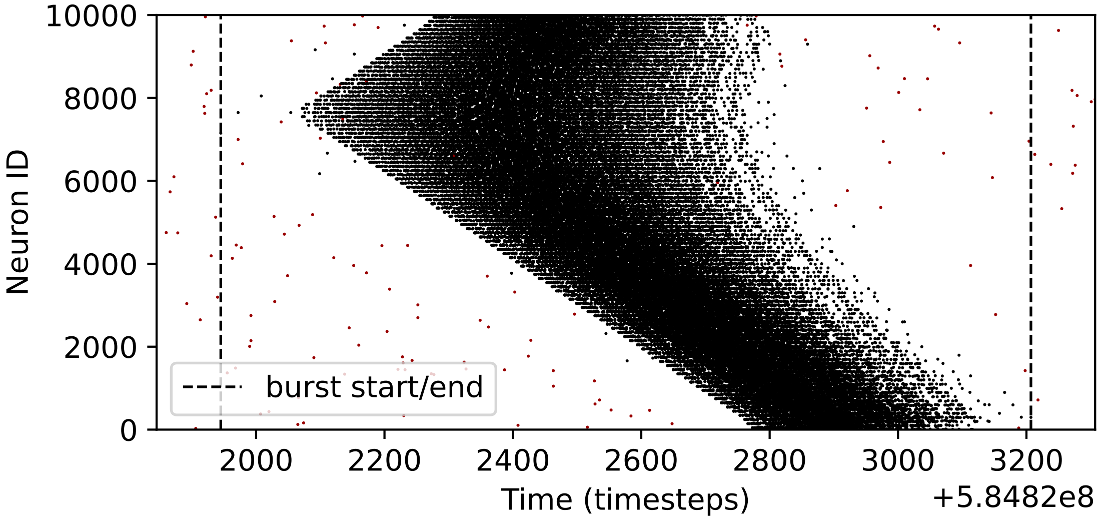
Alongside this increase, mid-sized avalanches ($16 \leq \mathit{avalSize} \leq 1000$), which were detectable under the $\tau = 1.5$, $r = 8$ constraints (Fig. 4.1), disappear when the temporal window is expanded to 50 timesteps (Fig. 4.2). This simultaneous increase in burst size/duration and the disappearance of mid-sized avalanches raises a key question: are the mid-sized avalanches genuine independent events, or are they components of whole-network bursts?

To investigate this further, we employed two complementary approaches: (1) temporal proximity analysis and (2) burst visualization.

[1] The duration of a burst is defined as the total time for which it recorded heightened spike activity i.e. $\mathit{endTime} - \mathit{startTime}$, while the size of a burst is defined as the number of spikes fired in that duration.



(a) Burst identified using the stricter window $\tau = 1.5$, $r = 8$. Dotted blue lines indicate the ±50 timestep window around the burst boundaries.



(b) Same burst identified using the more lenient window $\tau = 50$, $r = 8$. The burst boundaries expand, incorporating surrounding spikes.

Figure 5.1: Raster plots of an isolated burst. Each dot corresponds to a spike, with its horizontal position indicating spike time and vertical position indicating the spiking neuron's ID. Black dots represent burst spikes, while red dots denote all non-burst spikes, including noise. Dashed black lines mark the identified burst start and end times.

5.1 Temporal Proximity

Fig. 5.1a shows a raster plot of system activity during an isolated burst identified using the $\tau = 1.5$, $r = 8$ constraint. Black dots represent spikes classified as part of the burst, while red dots denote all non-burst spikes that may or may not be a part of other avalanches. Noticeably, several spikes occur within 50 timesteps before burst onset and after burst termination. These are not included in the burst under the tighter constraint, but we see that they become part of the burst when a more lenient temporal threshold is used, as shown in Fig. 5.1b.[2]

However, it remains unclear whether the newly added spikes are part of mid sized avalanches. To investigate this, we analyzed all mid-sized avalanches and computed their temporal distance from the nearest burst, as shown in Fig. 5.2. We found that virtually all mid-sized avalanches occur within 50 timesteps before burst initiation or after burst termination.

Notably, although a few mid-sized avalanches initially fall outside the 50-timestep window, they are eventually incorporated into bursts through transitive inclusion: as bursts absorb nearby avalanches, their temporal boundaries expand, bringing additional avalanches within range for merging.

5.2 Burst Visualization

In addition, high-resolution visualization of individual spike activity provided definitive evidence for the burst integration hypothesis. Fig. 5.3 shows that bursts propagate as coordinated wavefronts across the neural network. The interaction of the wavefront with the boundaries of the 100×100 simulation grid leads to localized clusters of activity, as shown in Fig. 5.4. Under restrictive temporal constraints, parts of the wavefront get detached and are

[2] Since the 10,000 neurons are arranged in a 100×100 grid, it's important to note that spatial contiguity in Figures 5.1a and 5.1b is not representative of actual spatial relationships in the simulation grid. For example, neuron IDs 100 and 101 appear adjacent in the plots mentioned above, but are located on the rightmost and leftmost edges of the simulation grid, respectively.

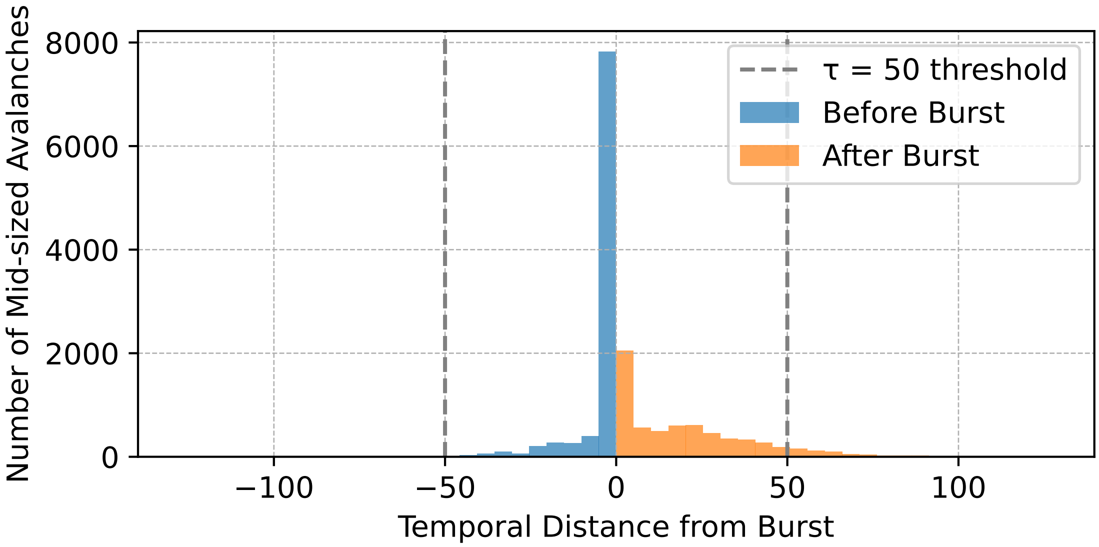


Figure 5.2: Temporal proximity of mid-sized avalanches to the start/end times of bursts.

misclassified as independent mid-sized avalanches; however, the expanded temporal window correctly identifies them as components of the originating burst.

These findings demonstrate that all mid-sized avalanches are artifacts of temporal windowing rather than independent network events.

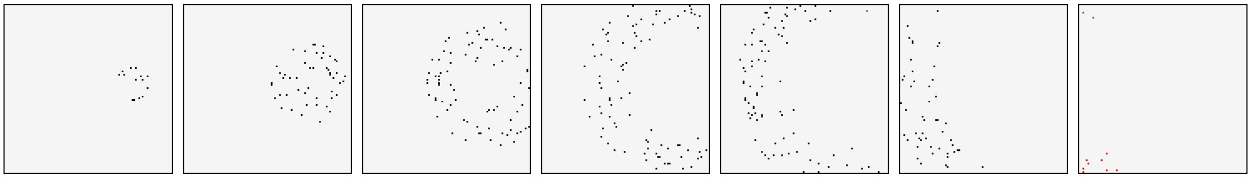


Figure 5.3: Evolution of a single burst event from start to end, ordered left to right in a 100×100 pixel-grid. Adjacent images differ by 130 timesteps (13ms). Each pixel represents a neuron in the 100×100 pixel-grid, color-coded according to activity—black dots represent spikes that were classified as being part of a burst, while red dots (visible in the last frame) represent spikes that were classified as part of a mid-sized avalanche.

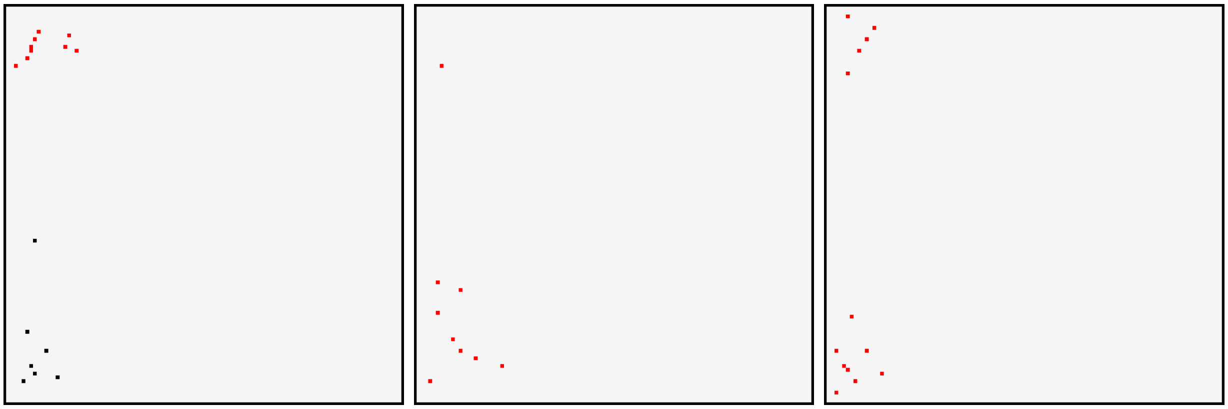


Figure 5.4: Trailing burst activity as the wavefront propagates to the edges of the simulation grid. Red dots indicate spikes belonging to mid-sized avalanches, while black dots denote burst-associated spikes.

Chapter 6

DISCUSSION

Most avalanche analyses in the literature focus on systems responding to external stimuli, where all resulting activity can be clearly attributed to avalanche events. In contrast, our neuronal simulations operate autonomously, driven by a combination of excitatory and endogenously firing neurons, creating a more complex analytical challenge. Without external stimuli to delineate avalanches, we must infer avalanche events from ongoing network activity while distinguishing them from stochastic background noise. Because direct causal relationships between spikes are difficult to establish in this setting, we apply spatiotemporal constraints to support avalanche inference.

6.1 Power-Law Relationships

The addition of spatial constraints ($\tau = 1.5, r = 8$) reduced detected avalanches from 6,340,187 to 473,509 while improving the power-law fit (R^2 from 0.937 to 0.953). With a more informed spatiotemporal constraint of ($\tau = 50, r = 8$), the power-law relationship manifests with an exponent of $\beta = -7.29$, where all mid-sized avalanches disappear. Although this yields a higher quality of fit ($R^2 = 0.955$), slope and fit metrics can no longer be trusted as indicators of confidence, since all meaningful mid-sized avalanches were later revealed to be part of burst activity.

6.2 Mid-Sized Avalanche Disappearance

The systematic increase in measured burst size and duration with expanded temporal windows prompted further investigation into whether mid-sized avalanches were being absorbed into bursts. Temporal proximity analysis revealed that the mid-sized avalanches, as identified

Table 6.1: Performance Benchmarking of Spatiotemporal Avalanche Detection Algorithm

Dataset Size (# of Spikes)	**Execution Time**
1.1 $\times 10^3$	0.006 seconds
5.7 $\times 10^4$	2.46 seconds
4.8 $\times 10^5$	19.4 seconds
5.2 $\times 10^6$	209 seconds (3.5 min)
1.7 $\times 10^8$ (complete dataset)	7,423 seconds (2.1 hours)

by the $\tau = 1.5, r = 8$ spatiotemporal constraint, consistently appeared within 50 timesteps of burst initiation or termination. This observation provided strong evidence that these events represent burst-related activity rather than independent avalanches.

High-resolution visualizations supported this interpretation, showing burst propagation as wavefronts that become fragmented at the edges of the simulation grid. Under restrictive temporal constraints, this fragmented activity was being misclassified as separate avalanches.

6.3 Algorithm Performance

A key focus of this research was the development of an *efficient* spatiotemporal clustering algorithm. Previous implementations of the algorithm by INL lab members required tens of hours to days for processing the complete dataset. Our performance benchmarks given in Table 6.1 demonstrate significant improvements in this regard:

Chapter 7
FUTURE WORK

A critical unresolved question concerns whether observed avalanche phenomena represent genuine emergent network behavior or artifacts of endogenously active neurons. 10% of the neurons in our simulation were set up to fire endogenously. Given that, future analysis can apply the spatiotemporal classification algorithm while excluding all spike events generated by these endogenous neurons. This approach would isolate avalanche activity arising purely from network connectivity and synaptic propagation, distinguishing it from activity driven by stochastic neuron activation. Comparing avalanche distributions with and without endogenous contributions would establish whether the observed power-law relationships reflect genuine self-organized criticality or represent statistical artifacts of autonomous neural activity.

The specific mechanisms that trigger burst initiation remain unexplored. Future research can focus on the systemic analysis of avalanche patterns in the time before burst onset to reveal characteristic signatures that predict burst formation. Understanding these pre-burst dynamics would provide insights into the network conditions necessary for large-scale burst activity.

BIBLIOGRAPHY

[1] LF Abbott and R Rohrkemper. A simple growth model constructs critical avalanche networks. *Progress in brain research*, 165:13–19, 2007.

[2] Per Bak, Chao Tang, and Kurt Wiesenfeld. Self-organized criticality: An explanation of the 1/f noise. *Physical review letters*, 59(4):381, 1987.

[3] Per Bak, Chao Tang, and Kurt Wiesenfeld. Self-organized criticality. *Physical review A*, 38(1):364, 1988.

[4] John M Beggs and Dietmar Plenz. Neuronal avalanches in neocortical circuits. *Journal of neuroscience*, 23(35):11167–11177, 2003.

[5] Lik-Chun Chan, Tsz-Fung Kok, and Emily SC Ching. Emergence of a dynamical state of coherent bursting with power-law distributed avalanches from collective stochastic dynamics of adaptive neurons. *PRX Life*, 3(1):013013, 2025.

[6] Dante R Chialvo. Emergent complex neural dynamics. *Nature physics*, 6(10):744–750, 2010.

[7] Taras A Gritsun, Joost le Feber, and Wim LC Rutten. Growth dynamics explain the development of spatiotemporal burst activity of young cultured neuronal networks in detail. *PLoS ONE*, 7(9), 2012.

[8] Guenter W Gross. Simultaneous single unit recording in vitro with a photoetched laser deinsulated gold multimicroelectrode surface. *IEEE Transactions on Biomedical Engineering*, (5):273–279, 1979.

[9] Beno Gutenberg and Charles Richter. *Seismicity of the Earth*, volume 34. Geological Society of America, 1941.

[10] Theodore Edward Harris et al. *The theory of branching processes*, volume 6. Springer Berlin, 1963.

[11] Eugene M Izhikevich, Niraj S Desai, Elisabeth C Walcott, and Frank C Hoppensteadt. Bursts as a unit of neural information: selective communication via resonance. *Trends in neurosciences*, 26(3):161–167, 2003.

[12] Fumitaka Kawasaki and Michael Stiber. A simple model of cortical culture growth: burst property dependence on network composition and activity. *Biological cybernetics*, 108:423–443, 2014.

[13] Jewel YunHsuan Lee, Michael Stiber, and Dong Si. Machine learning of spatiotemporal bursting behavior in developing neural networks. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 348–351. IEEE, 2018.

[14] John E Lisman. Bursts as a unit of neural information: making unreliable synapses reliable. *Trends in neurosciences*, 20(1):38–43, 1997.

[15] Hidetoshi Nishimori and Gerardo Ortiz. *Elements of phase transitions and critical phenomena*. Oup Oxford, 2010.

[16] Jerome Pine. Recording action potentials from cultured neurons with extracellular microcircuit electrodes. *Journal of neuroscience methods*, 2(1):19–31, 1980.

[17] Michael Stiber, Fumitaka Kawasaki, Delmar B Davis, Hazeline U Asuncion, Jewel Yun-Hsuan Lee, and Destiny Boyer. Braingrid+ workbench: High-performance/high-quality neural simulation. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2469–2476. IEEE, 2017.

[18] James Stirling. *Methodus differentialis: sive tractatus de summatione et interpolatione serierum infinitarum*. 1730.

[19] Christian Tetzlaff, Samora Okujeni, Ulrich Egert, Florentin Wörgötter, and Markus Butz. Self-organized criticality in developing neuronal networks. *PLoS computational biology*, 6(12):e1001013, 2010.

[20] CA Thomas Jr, PA Springer, GE Loeb, Y Berwald-Netter, and LM Okun. A miniature microelectrode array to monitor the bioelectric activity of cultured cells. *Experimental cell research*, 74(1):61–66, 1972.

[21] Jonathan Touboul and Alain Destexhe. Can power-law scaling and neuronal avalanches arise from stochastic dynamics? *PloS one*, 5(2):e8982, 2010.

[22] Jaap Van Pelt, Pieter S Wolters, Michael A Corner, Wim LC Rutten, and Ger JA Ramakers. Long-term characterization of firing dynamics of spontaneous bursts in cultured neural networks. *IEEE Transactions on Biomedical Engineering*, 51(11):2051–2062, 2004.

[23] Yuichiro Yada, Takeshi Mita, Akihiro Sanada, Ryuichi Yano, Ryohei Kanzaki, Douglas J Bakkum, Andreas Hierlemann, and Hirokazu Takahashi. Development of neural population activity toward self-organized criticality. *Neuroscience*, 343:55–65, 2017.