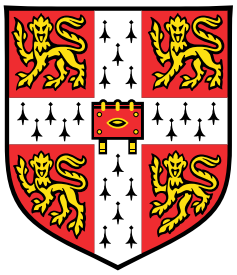
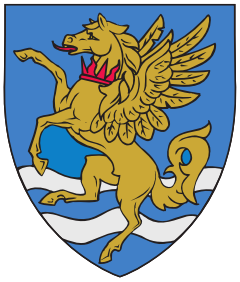


Oscillatory Spiking Circuits

A biologically plausible architecture for fast and reliable spiking computations



Thomas Simon Johannes Burger

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Robinson College

January 2024

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the preface and specified in the text. It is not substantially the same as any work that has already been submitted, or, is being concurrently submitted, for any degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee

Thomas Simon Johannes Burger
January 2024

Abstract

Oscillatory Spiking Circuits:
A biologically plausible architecture for fast and reliable spiking computations
Thomas S. J. Burger

The functions carried out by the brain are completed in a large part by dynamically complicated neurons which transmit information in discrete pulses, called spikes. The brain uses these to compute in a (usually) reliable fashion, and can do so rapidly. How to construct Spiking Neural Networks (SNNs) that achieve similar performance is still unknown. Here, we present an architecture for spiking neural networks that operate efficiently (with a low number of spikes), rapidly, and robustly, using biologically plausible components and operating within biological constraints.

At the heart lie neuronal oscillations which are produced by recurrent excitation and inhibition, called PING. The oscillations organise the spikes in time, putting the circuit in a binary regime, with each neuron having an integrating time window that can accommodate at most one spike from each upstream neuron.

Firstly, we investigate how naive approaches to construct SNNs can fail when confronted with spike timing jitter. We then propose a computation role for dendritic action potentials with long time constants. Such dendritic spikes exhibit long plateau potentials which can last tens of milliseconds, outliving somatic spikes by an order of magnitude. We propose that this long time constant allow for the reliable integration of asynchronous inputs.

Then we turn our attention to PING rhythms. We briefly look how they arise and show that these rhythms remain reliable in the face of disturbances and parameter uncertainty, and introduce a tool to investigate feedback on the level of the network: the population voltage clamp. Using this, we demonstrate that PING is reliable due to slow positive feedback arising from synaptic interactions.

Finally, we make the case that PING circuits can indeed support a binary computation. After examining why gradient-based methods can struggle in the training of spiking systems, we introduce a new method of training biological neural networks, which we call teacher-forcing.

Table of contents

List of figures **ix**

List of tables **xi**

1 Introduction **1**

2 Effects of timing jitter on spiking computations **9**
2.1 Timing jitter in naïve SNNs 9
2.1.1 Timing jitter in single neurons 9
2.1.2 Implications for network performance 11
2.2 Single neuron mechanisms for reducing jitter 20
2.2.1 Abstract model of dendritic transients 22
2.2.2 Dendrites transients aid integration of asynchronous spikes 26
2.2.3 Active dendrites confer robustness in spiking computations 27
2.3 Discussion 31
2.4 Methods 36
2.4.1 Hodgkin-Huxley Neuron 36
2.4.2 ANN classification 37
2.4.3 Leaky-Integrate and Hold Model 37
2.4.4 Neuron with asynchronous inputs 38
2.4.5 Binary network 38

3 Network mechanisms for reducing jitter **41**
3.1 PING rhythms 41
3.2 Origins of PING 45
3.2.1 Computation of mean-field E-I current 48
3.2.2 Dynamic input conductance 55
3.3 Discussion 61
3.4 Methods 66

3.4.1 AMPA / $GABA_A$ currents 66
3.4.2 Necessary conditions for PING 66
3.4.3 Voltage clamp 67

4 A framework for reliable SNNs **69**
4.1 BNN interpretation of PING rhythms 69
4.2 PING circuits can compute 71
4.3 Training SNNs using the teacher forcing algorithm 73
4.3.1 Feedback sensitivity 73
4.3.2 Surrogate gradients and conductances 78
4.3.3 Teacher forcing 83
4.4 Discussion 91
4.5 Methods 92
4.6 PING BNN 92
4.6.1 Feedback sensitivities 92
4.6.2 Surrogate gradients 93
4.6.3 MQIF model 93
4.6.4 Teacher forcing 96

5 Discussion **99**
5.1 Recapitulation 100
5.2 Future work 101
5.3 Final remarks 103

References **105**

List of figures

1.1 Example of an action potential 2

2.1 Effects of spike-timing jitter for single neuron 12
2.2 Toy classification problem 13
2.3 Performance of stochastic SNN 16
2.4 LIF activation function 18
2.5 LIF-based SNN performance 19
2.6 Dendritic dependent action currents 21
2.7 NMDA currents in data, a biophysical model, and an abstract model 23
2.8 Dendritic transients reduce uncertainty due to timing jitter 28
2.9 Active dendrites help make an SNN robust to timing jitter 32

3.1 PING E–I architecture 42
3.2 Example of E–I PING 44
3.3 Conditions for PING rhythm 46
3.4 Population voltage clamp example 51
3.5 PING mean-field gating steady-state and time-constant functions. 52
3.6 PING with I_{EI} mean-field approximation in open loop. 53
3.7 PING with I_{EI} mean-field approximation in closed loop. 54
3.8 Decomposition of ionic currents in separated timescales. 57
3.9 PING mean-field I_{EI} conductance in several timescales 60
3.10 Diminished synchrony in PING with LIF inhibitory neurons 62
3.11 LIF PING population voltage clamp 63

4.1 Illustration of how a PING networks mimics a BNN 70
4.2 Coherence and participation in PING 72
4.3 Mutual information of inputs and outputs of multi-layered PING circuit . . 74
4.4 Block diagram of feedback in the dynamics of a linearised neuron 76
4.5 Amplification of inputs in a linearised conductance neuron model 77

4.6 Example of training SNN with surrogate gradients 80
4.7 Surrogate gradient training in MQIF network 82
4.8 Amplification of inputs in the MQIF and LIF model 83
4.9 Sketch of the network trained with teacher forcing 85
4.10 Illustration of the construction of an artificial voltage trace based on a binary word an a timeperiod as inputs. The binary vector on the left is implemented as the target spiking output of a neuron through the artificial voltage shown on the right. The inlays at the top represent the ‘spike’ and ‘no spike’ time series respectively which were concatenated to form the overall target voltage trace. 86
4.11 Teacher forcing result of on/off task 89
4.12 Teacher forcing result of diagonalisation task 90
4.13 Example of input used in surrogate gradient training 94

List of tables

2.1 Ion channel densities of morphologically detailed, biophysical model 24
2.2 Hodgkin-Huxley model parameter values 36
2.3 LIH model parameter values 38

3.1 Default values for simulations with AMPA currents 66
3.2 Default values for simulations with $GABA_A$ currents 66

4.1 MQIF model parameter values 94

Chapter 1

Introduction

The brain is an impressive machine. It has helped to give us things such as calculus, the *Requiem*, *het melkmeisje*, and helped to put people on the moon. It is natural that people are fascinated by the deceptively simple looking squishy blob inside our skulls, which, upon closer inspection, reveals billions of neurons of hundreds of types with complex electric behaviour, connected via trillions of synapses, supported by countless chemical pathways for information processing and maintenance, amongst other components.

It has become a habit to describe this blob with references to the most spectacular technology known to the era. Greeks in antiquity used their pneumatic water technologies, such as fountains, to describe the concept of the *pneuma* (soul) [161]. During the Renaissance the technology of mechanical automata spread; a ticking clockwork can be heard underneath the thesis that man is a machine as defended by de La Mettrie [114]. In the latter half of the 19th century, this metaphor was superseded by electronic communication, as Helmholtz compared the nervous system to a series of transmitters and receivers connected by telegraphic wires [70]. Seen in this light, it is expected that today's thinking of the brain is coloured by that artefact which has become commonplace in our everyday life: the digital computer. This metaphor for the brain was introduced by McCulloch and Pitts [112], who proposed that neural function implemented propositional logic, an idea that John von Neumann [**neumannComputerBrain2012**] and Norbert Wiener [168] delved into later. Indeed, these developments led to the brain being viewed through a lens of logic gates and algorithms, which led to the word 'computation' being used as a metaphor for the brain's functioning.

This metaphor for the brain has been reinforced over the past decade or so by computing systems acquiring an increasingly impressive array of skills, such as image generation and mimicking human chatter. But this technological progress has been enabled by using Artificial Neural Nets (ANNs), which is the brain-inspired computing architecture introduced by McCulloch and Pitts. To complete the circle, ANNs are now used in neuroscience, both

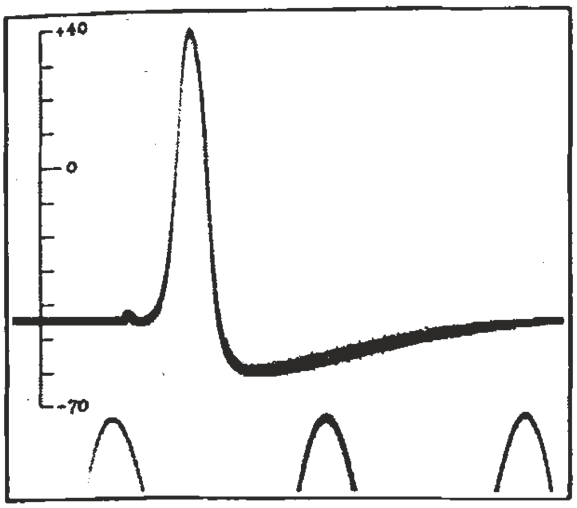


Fig. 1.1 An example of an action potential or *spike* as recorded in the giant axon of a squid (*Loligo forbesi*). The vertical scale indicates the potential on the inside of the membrane, the oscillation below the voltage trace has a frequency of 500 Hz. Reproduced from [76].

as a tool to aid experimentalists, and as computational models of the functions performed by neural circuits [85, 108, 129, 111].

But, although inspired by biological facts, ANNs share few similarities with biological circuits. ANNs are mathematical abstractions, where every neuron or 'unit' performs a few elementary mathematical operations, and need not have a direct relationship with the hardware they are implemented on. Biological neural circuits, on the other hand, are defined by their components. The brain is a physical device that fulfills a specific set of functions, and obeys engineering principles. The brain computes with chemistry when speed over large distances is not crucial, and for speed over distance it computes electrically [89, 150]. The basis for this rapid long-distance signalling in larger brains is the sharp, brief, energy-intensive action potential which is commonly referred to as a *spike*, an example of which is shown in Figure 1.1. It is a pulse in the voltage across the cell membrane of a neuron, which is denoted by the symbol V.

In this thesis we are interested in spike-based computations, which is why the membrane potential V is the primary physical substrate we focus on. We have the benefits that we understand the biophysics that regulate the voltage well [77, 75], and it is a signal relatively easily measured in electrophysiology. The question we seek to answer is that of implementation: how does the brain carry out its functions using spikes, and which components allow it to do this? It turns out that this is a difficult question, as we have not yet understood the engineering principles behind biological Spiking Neural Networks (SNNs). They are difficult to work with for several reasons. Firstly, spikes are a binary signal, which can make them

bandwidth limiting compared to analogue signals when used for simple computations [115]. Secondly, the workhorse of training ANNs, namely stochastic gradient descent, does not work straightforwardly on spiking systems, as their gradients tend to explode or vanish [117]. And lastly the subthreshold behaviour of biologically plausible spiking systems is complex, which makes engineering them difficult. To begin with finding an answer of how to work with spikes, we must ask ourselves what spikes are, and what the evidence tells us about their meaning.

Spikes are a digital phenomenon: when a nerve cell is stimulated with sufficient intensity, it produces an action potential close to its soma, which then travels down its axon at constant amplitude and velocity. Action potentials have a stereotyped shape (which depends on the cell type, not on the neuron's input that caused it to spike), and are an all-or-none phenomenon. That is, if the membrane potential V of a neuron fails to exceed some critical value, no spike is fired. But if it exceeds this value, called the firing threshold, an action potential is produced. However, not all neurons produce action potentials: cells that communicate over small distances, e.g. neurons in the retina or in small organisms such as *C. elegans*, do not need action potentials and instead transmit information via analogue graded potentials [131].

So why did action potentials evolve in the nervous system? One straightforward answer is that it allows for long-distance communication in animals with larger bodies. If axons and dendrites were cables without active elements, the passive conduction would lead to an exponential decay of the signal with a length scale of $\sim 1\,\mathrm{mm}$ [78, 127]. As a consequence, neurons evolved an active mechanism, namely action potentials, to quickly send information with little loss over distance.

But this explanation is not entirely satisfactory. Transmitting information in a pulsatile code with spikes is qualitatively different from using graded signals when sending analogue information, which the nervous system still doubtlessly does. So how do biological SNNs encode information in their spiking output, called the spike train? The classical view is that of *rate coding*, where the firing rates, i.e. the frequency with which spikes are emitted (multiple precise definitions are possible, see e.g. [130, 28]), is the primary quantity with which the nervous system operates [5, 4, 93, 12]. The firing rate is an analogue quantity, so in this view the spike is reduced to a mechanism which developed to overcome challenges in long-distance communication in neurons, but was otherwise an inconvenient development which made the nervous system develop a digital to analogue, and vice versa, conversion mechanism.

Other neural codes are gathered under the umbrella term of *temporal coding*, where the (relative) timing of spikes carries information. In this view, the introduction of spikes in the nervous system added more sophistication than only allowing for fast communication over

distances: the fact that spikes are an event that have a specific timing is exploited for the neural code.

Population codes are codes where the atom of information is not the activity of any one neuron considered independently, but is instead the activity of a collection of neurons, and can be both rate based and time based, depending on the details.

A considerable amount of debate has gone into the question of which neural code is applicable when and where in the brain. The debate is not settled yet, with both theoretical and experimental arguments having been advanced by both sides.

Firing rate models have the advantage they are easy to work with: they bridge the gap between ANNs, which operate in analogue fashion, and SNNs. The firing rate has been shown to be a correlate with biologically relevant variables: for example, it is a predictor of features in animals' behaviour, such as discrimination tasks [165, 13, 175], and it has been shown to vary smoothly with perceptual variables [133, 116]. And a common argument in favour of a rate based interpretation, or more precisely against timing based codes, is the observation that neural activity appears random. For example, the timing and number of spikes fired by neurons in response to a specific stimulus is variable [157, 175, 147, 143]. This has lead to the modelling of neural activity as a random Poisson point process, sometimes with high accuracy [122, 11].

Moreover, by averaging spike trains of neurons over multiple trials in which animals presented with the same stimulus in each trial, it was found that this averaged firing rate correlated strongly with some features of the stimulus. These trial-averaged firing rates are called tuning curves, and have been very successful in describing how information from a broad scope of senses is relayed to areas of the brain for processing [79, 130, 45]. Contrast this fact with the apparent randomness of neural activity: on the one hand, neuronal firing patterns are very irregular, but on the other hand we can retrieve the 'data' a neuron transmits successfully when the activity is averaged over many trials, and a regular pattern eventually emerges. A rate code, according to this argument, would deal with variability through such an averaging procedure [98], and it would therefore be very robust to changes in the properties of the neural circuit performing the computation. In this interpretation, individual spikes are not that relevant, and therefore little information is lost by the averaging procedure.

The contrasting view, and the one we take in this thesis, is that individual spikes do in fact matter.

The first argument that speaks in favour of this are the data from psychophysical experiments. Experiments showed that monkeys can very accurately complete visual recognition tasks in about 100 ms [123, 132]. There is a population of neurons in their temporal lobe that is selective for facial features [123], with reported latencies of 100 ms to 140 ms, The

shortest known anatomical pathway consists of ~ 10 stages, whilst typical firing rates of neurons along this path are well below 100 Hz [154], leaving on average enough time for 1–2 spikes per neuron in the pathway. Griffin et al. [66] showed that discovered that bats employ echolocation to hunt for insects in the dark, emitting ultrasonic signals and listening to the echoes to locate prey. Griffin showed that the signals that the little brown bat receives are short, about 0.6 ms to 1 ms. Galazyuk and Feng [58] confirmed that neurons in several layers of the brain of the little brown bat individually fired only a handful of spikes within a timespan of 500 ms. O'Keefe and Recce [118] demonstrated that rats can do complex navigational tasks with a limited number of spikes, even though the rat hippocampus receives inputs from areas of the brain which process sensory inputs. In the cortex *in vivo* firing rates are small, on the order of 1 Hz [145]. In these cases, on average a neuron in the pathways can fire about one or two spikes, in order to complete complex tasks: an insufficient number to build up a firing rate estimate, unless a population code is used.

Secondly, neurons, when examined *in vitro*, are quite reliable. Calvin and Stevens [39] found that variability in the statistics of spike timings was explained by noisy inputs to the neuron, with neurons themselves not introducing uncertainty. And in the experiments of Bryant and Segundo [30] and Mainen and Sejnowski [104] it was shown that neurons can reproduce reliable spike sequences if subjected to a time varying input with fast enough fluctuations. Furthermore, the timing of action potentials of neurons *in vivo* is also reliable with respect to the firing times of other neurons in the brain [151, 2, 71]. Given that neurons therefore seem to be reliable, assuming they fire randomly, and therefore require a rate code, appears a strong assumption. Barlow [12] puts it nicely:

> Individual nerve cells were formerly thought to be unreliable, idiosyncratic, and incapable of performing complex tasks without acting in concert and thus overcoming their individual errors. This was quite wrong, and we now realise their apparently erratic behaviour was caused by our ignorance, not the neuron's incompetence. Thus, we gain support from this neuropsychical comparison for the concept of a neuron as a reliable element capable of performing a responsible role in our mental life.

Lastly, a firing rate code requires, by definition, a high number of spikes. But spikes are energetically costly, consuming about 50% of the brain's energy [9, 140]. To be energy efficient with spikes, a time based code must be used over a rate based code [44]. Moreover, a consequence of information theory is that any code using fewer spikes conveys more information per spike [144, 95], and information rates increase sublinearly with spike rates. Consequently, the brain evolved in a multitude of ways to maximise information and energy efficiency, that is, to limit spike usage and keep firing rates low. The *thalamus* in mammalian

brains, for instance, is used to reduce firing rates before relaying messages to the cortex [14, 150]. Given the benefits of low spike rates and the mechanisms that evolved to achieve them assuming a spike efficient code is reasonable.

The experimental data and theoretical arguments listed above shows that spikes contribute more than only allowing for fast, long-distance signalling. But all the benefits that can be derived from using spike based computations over using rate codes rely on using some kind of timing based code [103, 44]. This implies that there is a degree of coordination in time between the activity of neurons; which for most conceivable time based codes implies some sort of clock to provide a degree of synchrony.

Neural oscillations are an important mechanism for regulating neural activity and can provide such a clock. They represent the synchronised electrical activity of large neuronal populations, across a frequency band of 0.1 Hz to 100 Hz, which can be large enough to be observed even through the skull with an EEG [27], and they are widely observed across many species [37]. The mechanisms underlying the rhythmicity are diverse, and span from the dynamics of ion channels and synapses to patterns of connectivity within neural circuits [163, 167].

We already mentioned that in many instances neurons have to receive spikes and process them to make a firing decision in as little as 10 ms. This restriction points us towards the *gamma* frequency band of neuronal oscillations: neural rhythms with frequencies of up to 100 Hz. Gamma oscillations have the benefit that they aid neurons in the reliable firing of action potentials with precise timing [30, 104, 68], which makes them useful from the implementation point of view.

Our objective is to construct a biologically plausible SNN that faithfully reflects the known principles of spike based computations in the brain and adheres to general biological constraints. In pursuing this goal, we aim to leverage the phenomenon of neural oscillations, as they are a sensible starting point for the biophysical underpinning of timing based computations in the neuronal circuits. In doing so we hope to help bridge the gap in our understanding of the detailed biophysics of single neurons and the abstract computational models of ANNs (which, as mentioned, are believed to reflect what the brain does, yet it is unknown how the brain does so!). As an aside, we mention that this question matters for another reason: understanding what the biological implementations of SNNs are has applications in neuromorphic engineering, which aims to design artificial systems that are equal to the brain in terms of efficiency and functionality [113].

We use theoretical tools to help answer these questions. In chapter 2 we discuss the difficulty of performing spike based computations given some uncertainty in incoming spikes, which all biological neural circuits must deal with. We discuss how active dendrites are a

component that furnish the neurons they are added to with an additional long time constant, which ameliorates the problem of integrating asynchronous spikes. In chapter 3, we move up a level from individual neurons to the neural circuit level, and treat neural oscillations as a mechanism for reducing spike timing jitter and as the basis for building reliable biologically plausible SNNs, and investigates the biophysics of gamma oscillations. chapter 4 gives a framework for building reliable SNNs when building upon the results for chapter 3. The difficulties of training biological SNNs are explored, and an alternative training algorithm is presented. Finally, we summarise and discuss our findings, and conclude with further work that is yet to be done to build upon these results.

Chapter 2

Effects of timing jitter on spiking computations

2.1 Difficulties with timing jitter in naïve SNNs

2.1.1 Effects of timing jitter on single neurons

In this chapter we will illustrate the difficulties that SNNs can face when dealing with jitter in the spike-times, that is, how much does uncertainty in the timing of input or output spikes affect the performance of spiking networks. This jitter is hard to avoid for a biophysical network that is subject to uncertainty and noise. Such noise and variability in the nervous system has many sources. Firstly, the nervous system is heterogeneous, with the components of the neural circuitry always changing and reconfiguring within an individual [110, 109]. Other factors, such as the inherent noisiness of the production and degradation machinery of proteins, conduction delays, and noise in synapses make perfect synchrony impossible in circuits in the brain [53].

How much of a problem is this? To investigate this question, we first zoom in on individual neurons, and see how the output of a single neuron is affected by uncertainty in its inputs.

We begin with using a *conductance-based* point neuron model. A point neuron model is a model which ignores any spatial variations or geometry in the neuron, and treats the ion channels and membrane potential to be the same everywhere in the neuron, thereby reducing it to a point. The membrane is modelled as a capacitance which integrates the membrane current $I_m(V,t)$ flowing through the neuron membrane that is the result of a host of different types of ion channels opening and closing. Usually an external applied current I_{app} is included also. The fraction of channels of ion i that is open at a given moment is modelled

by a conductance $g_i(t,V)$. From Ohm's law we know that the current flowing through an ion channel is the product of this conductance with a voltage difference of the membrane potential V and the equilibrium potential E_i:

$$\begin{aligned} C\dot{V} &= -I_m(t,V) + I_{\text{app}} \\ &= -\sum_i g_i(t,V)\,(V - E_i) + I_{\text{app}}. \end{aligned} \tag{2.1}$$

The dynamics of the ion channels are usually modelled by introducing gating variables, which depend on the history of the membrane potential:

$$\begin{aligned} g_i(t,V) &= \bar{g}_i x_{i1}^a x_{i2}^b \\ \tau_{x_{ij}}(V)\dot{x}_{ij} &= x_{ij,\infty}(V) - x_{ij} \end{aligned} \tag{2.2}$$

for a maximal conductance $\bar{g}_i$, some functions $\tau_{x_{ij}}$ and $x_{ij,\infty}(V)$, and exponents a and b. Here x_{i1} denotes the activation variable of channel i, meaning that $x_{i1,\infty}(V)$ is an increasing function of V, and x_{i2} is the inactivation variable, which means that $x_{i2,\infty}(V)$ decreases with V.

We simulated a single-compartment Hodgkin-Huxley neuron model receiving input spikes that were jittered in time (Fig. 2.1). This model is the classical example of a conductance-based neuron model, with its dynamics given by

$$\begin{aligned} C\dot{V} &= -\underbrace{\left(\bar{g}_{Na} m^3 h (V - E_{Na}) + \bar{g}_K n^4 (V - E_K) + \bar{g}_l (V - E_l)\right)}_{I_m} + I_{\text{app}}(t) \\ \dot{m} &= (m_\infty(V) - m)/\tau_m(V) \\ \dot{h} &= (h_\infty(V) - h)/\tau_h(V) \\ \dot{n} &= (n_\infty(V) - n)/\tau_n(V), \end{aligned} \tag{2.3}$$

For the steady-state and time-constant function $x_\infty(V)$ and $\tau_x(V)$, and for the parameter values, see section 2.4.1.

This neuron was simulated with two classes of input spikes: the first ensemble of simulations had a neuron whose input spike volleys arrived in a narrow window of time, with jitter in the spike timing of order 1 ms ($t_i \sim \mathcal{N}(20\,\text{ms}, 1\,\text{ms})$, t_i being the time of input spike i), and the second ensemble had a neuron with timing jitter in input spikes of the order of 10 ms ($t_i \sim \mathcal{N}(20\,\text{ms}, 10\,\text{ms})$), where the neuron itself had a membrane capacitance of $C = 1\,\mu\text{F}/\text{cm}^2$.

When the input spikes arrive in a narrow time window, the neuron responds reliably, assuming that it otherwise remains constant in time. The standard deviation of the membrane

potential of a neuron receiving the input spikes with low timing jitter is $\sigma = 0.12$ (Figure 2.1B, left). The output spikes occur almost all in the same timebin (Figure 2.1C, left, width of timebins is 1 ms).

It is apparent that jittered input spikes lead to more variability in the membrane potential ($\sigma = 0.24$), and to a net decrease in overall depolarization (Fig. 2.1B, right). This, in turn, leads to the neuron producing output spikes at more erratic times, and to it failing to spike in many cases, leading to a failure rate of spiking of about 20% (Fig. 2.1C, right).

It's worth briefly mentioning that all of this can be fixed by increasing the total number of input spikes. As is shown in Fig. 2.1D (right), the coefficient of variation can be kept small for an arbitrary amount of jitter by increasing the number of spikes that the neuron receives. The relationship between the amount of jitter in the spiketimes and the number of inputs required to keep the coefficient of variation (the standard deviation over the mean) small is linear, and quickly gets into the hundreds of spikes necessary for a desired small coefficient of variation. This is not an energetically optimal solution, as spikes come at a cost: it is estimated that action potentials consume about half of the energy that the brain uses in total [9], so using more spikes means considerably increasing the energy budget of neural circuits. Finally, we remark that these results depend on the neuron model not having any spatial extent. Indeed, in section 2.4.3 we investigate how neurons with spatial dimensions can get around these issues.

2.1.2 Implications for network performance

The results in the above shows that uncertainty in spike timings introduces uncertainty in neuron behaviour – a completely unsurprising conclusion, of course. Why is it worth pointing this out nevertheless? As mentioned in the introduction, it is sometimes assumed that moving from rate-models to spiking models is easy. The above shows that this is actually not as simple as it might appear at a glance, unless one uses many inputs to balance out the uncertainty in the spike timings, for example by integrating over long periods of time or many spikes, e.g. by using a population code. In the following, we explore briefly which problems one can run into if one naively attempts to convert a rate-based ANN to an SNN

To start with such a conversion, one needs a working ANN first. Therefore, we trained a feedforward ANN to perform a simple toy task. For simplicity, we chose a classification task, as the high performance of ANNs on such tasks is well established. We created some data that the ANN will have to classify. One of the simplest classification tasks there is, is that of classifying Gaussian 'blobs'; points generated from different Gaussian distributions. Then the ANN has to learn to which distribution, or class, each point belongs. We will work with 2D points. Each point p_i^N, which is the ith point belonging to the Nth class, is generated

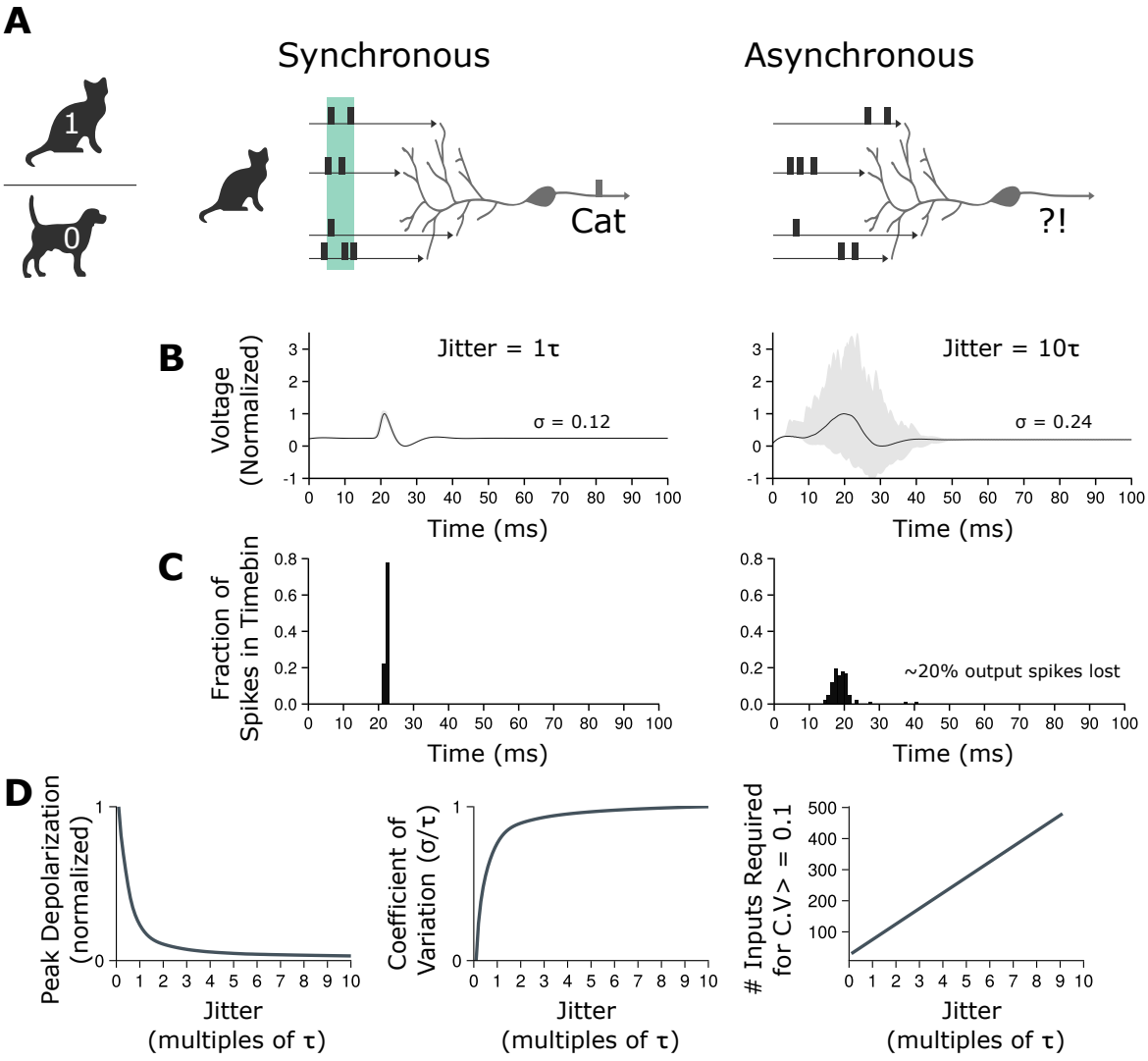


Fig. 2.1 The effects of timing jitter in the input spikes on the voltage of a neuron and the timing of its output spikes. **(A)** Neurons integrate input spikes and compare the outcome to a threshold, thereby deciding to spike or not. This comes down to performing a binary classification, which is depicted here as deciding between a cat or a dog. When the input spikes arrive in a small time window this can be done with a few of spikes (left). But when the spikes arrive in a wider time window, the output spiking becomes unreliable (right). **(B)** The voltage of a single-compartment Hodgkin-Huxley style neuron with membrane time constant τ, receiving input spikes with a timing jitter of 1τ (left), and 10τ (right). More jitter increases the variability in the membrane potential and reduces net depolarization. **(C)** The jitter in the input spikes leads to unreliability in neuron spiking. When spikes arrive in a very narrow time window, the neuron spikes reliably (left). But when the inputs are jittered, the neuron will often fail to spike, and if it does spike it will do so at variable times (right). **(D)** Peak depolarization decreases sub-linearly as a function of jitter (left), whereas the coefficient of variation increases super-linearly (middle). The number of input spikes required to keep the variation in the membrane potential small increases linearly, and quickly runs into the hundreds (right).

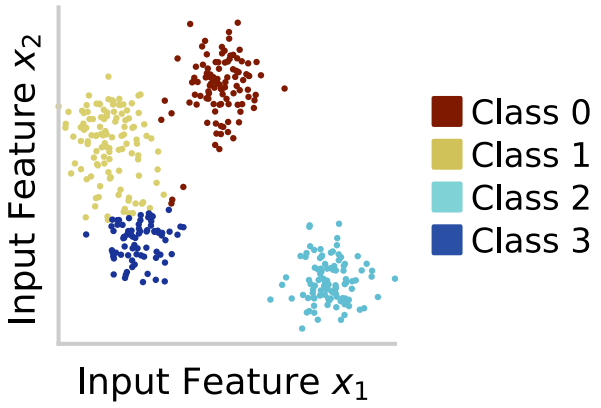


Fig. 2.2 Example of our toy classification problem, with the points belonging to four classes.

from a distribution $p_i^N \sim \mathcal{N}(\mu^N, \Sigma^N)$, where $\mu^N \in \mathbb{R}^2$ is the mean of its distribution, and $\Sigma^N \in \mathbb{R}^{2\times 2}$ is its covariance matrix. For the values of these parameters that were used to define the distributions, see section 2.4.2. An example of the classification task is shown in Figure 2.2.

First we trained an ANN on the data of this toy problem, with regular sigmoidal activation functions:

$$\sigma(x) = 1/(1+\exp(-x)). \tag{2.4}$$

The ANN needs to have at least one hidden layer, because this classification problem is not linear. All the ANN networks we discuss henceforth will have the same feedforward architecture. The input layer has two units, as the inputs to the network are the (x,y) coordinates of the points p_i^N. Then there are three hidden layers, with ten units in each. Finally, there is an output layer with four units. To give the correct answer, the Nth output unit must switch 'on' (bring its output close to one) for an input point p_i^N.

To train the network we used a straightforward supervised learning procedure. The network outputs $\hat{y}(x)$ were first parsed through the softmax function, so that the outputs of the network can be interpreted as probabilities, i.e. each output value lies between 0 and 1, and the outputs together sum to 1:

$$s(\hat{\mathbf{y}})_i = \frac{e^{\hat{y}_i}}{\sum_{j=1}^{K} e^{\hat{y}_j}}. \tag{2.5}$$

The resulting $s(\hat{\mathbf{y}})_i$ terms were then fed into the cross-entropy function,

$$H(\hat{\mathbf{y}}(\mathbf{x}), \mathbf{y}) = -\frac{1}{N}\sum_{i=1}^{B}\sum_{j=1}^{K} \log(s_j(\hat{\mathbf{y}}_i))\, y_{i,j}, \tag{2.6}$$

where $\hat{\mathbf{y}}_i$ is the network output in response to input $\mathbf{x}_i$ belonging to the batch of size B and $y_{i,j}$ the jth element of the corresponding target. We used a batch size of $B = 32$, for 100 points per class, and trained the model for 1000 epochs. Because of how simple this toy problem is, the ANN easily trains to around $99 - 100\%$ efficiency (where the difference depends on the randomness of the initialisation procedure).

We note that the only parameters that underwent training where the weights between the units. All biases were kept constant at -1. This choice was made so that the conversion to an SNN would be a bit more natural, as this choice forces the units to receive a net positive input equal to 1 before they switch 'on'. In an SNN this corresponds to a neuron requiring some positive input before spikes are produced. Note that the choice for which constant the bias equals is not important, and does not limit the computational power of the ANN provided it is sufficiently large, see 2.4.5 for details.

Stochastic SNN

The first conversion to an SNN that we attempted was to replace each ANN unit with a stochastically spiking unit, since this is one of the simplest descriptions of neuronal firing. Such models are popular because they capture the apparent stochasticity of neuronal firing, such as having a Fano factor that is almost one, and displaying a quenching of activity variability when stimuli are presented [65, 153, 41].

Such stochastic units fundamentally work by interpreting the 'rate' of the ANN units as a spiking probability of the spiking neurons in the SNN. We start with the layered, feedforward network of the ANN, and its trained parameters. The vector of output firing rates of layer $n+1$, which we denote as $\mathbf{x}^{n+1}$, is computed as a linear-nonlinear function of the firing rates of the previous layer, $\mathbf{x}^n$:

$$\mathbf{x}^{n+1} = \sigma(W^n \mathbf{x}^n - 1), \tag{2.7}$$

where W^n is the matrix of trained weights, and -1 is the vector of biases of the individual neurons in this layer. To move to continuous time, we model the dynamics as a first-order filter with time constant τ:

$$\tau \dot{\mathbf{x}}^{n+1}(t) = -\mathbf{x}^{n+1}(t) + \tau\sigma(W^n \mathbf{x}^n(t) - 1) \tag{2.8}$$

In order to simulate this, we switched to a discrete ODE solver method. We chose the exponential Euler method [40]. This leads to the following update rule for timestep $t+1$:

$$\begin{aligned} \mathbf{x}_{t+1}^{n+1} &= \alpha \mathbf{x}_t^{n+1} + (1-\alpha)\sigma(W^n \mathbf{x}_t^n - 1), \\ \alpha &= e^{-\Delta t/\tau}. \end{aligned} \tag{2.9}$$

The spiking itself we model as a Bernoulli process, in which each neuron emits a spike with probability $p = \gamma x$. This leads to the definition of the spike as a Kronecker (in discrete time) or Dirac δ (in continuous time) function. Here, we work in discrete time, and therefore use the Kronecker δ:

$$\delta_{tt_s} = \begin{cases} 1 \text{ if } t = t_s, \\ 0 \text{ otherwise.} \end{cases}$$

(Note that with Hodgkin-Huxley type models, as in Equation 2.3, the spike is an event that is continuous in time, and is therefore harder to formally capture in a mathematical equation.) The parameter γ controls the firing rate of the neuron, and can take values $0 \leq \gamma \leq 1$. We used a stepsize of $\Delta t = 1\,\text{ms}$, which means that the maximal firing rate possible is $\lambda_{\max} = 1000\gamma$. We also re-scale the weights, to preserve the average amount of input for all values of γ. This gives the update rule from timestep t to timestep $t+1$ of

$$\begin{aligned} y^{i,n} &\sim \text{Bernoulli}(p = \gamma x^{i,n}) \\ \mathbf{x}_{t+1}^n &= \alpha \mathbf{x}_t^{n+1} + (1-\alpha)\sigma\left(\tfrac{1}{\gamma} W^n \mathbf{y}_t^n - \vartheta\right) \\ \alpha &= e^{-\Delta t/\tau}. \end{aligned} \tag{2.10}$$

The inputs were kept analog, and were presented to the network for a duration of $T_{\text{stim}}/4$. We used a value of $T_{\text{stim}} = 60\,\text{ms}$. The input points p_i^N were first multiplied with W^1, which was then fed into the first hidden layer as a vector of two constant currents: $\mathbf{x}^1(t) = \left(W^1 p_i^N\right) H(t - T_{\text{stim}}/4)$ is the input to the first hidden layer for input i of class N. The rest of the network was given the rest of the period T_{stim} to settle on an answer.

The answer of the SNN was determined by counting which output neuron spiked the most for the interval in which the stimulus was presented. For a maximal firing rate of $\lambda_{\max} = 50\,\text{Hz}$ ($\gamma = 0.05$), the accuracy was 30%, or just above chance level, which is 25%. Overall the performance is very poor, as can be seen in Fig. 2.3A. When we increase the maximal firing rate to $400Hz$ ($\gamma = 0.4$), the ANN performance is retrieved (Fig. 2.3C). Likewise, if we use a population code, by replacing each spiking neuron with 50 copies of itself, we can achieve the same performance without resorting to unphysiological firing rates (Fig. 2.3D). This is in line with Fig.2.1D, right, where we showed that increasing the number

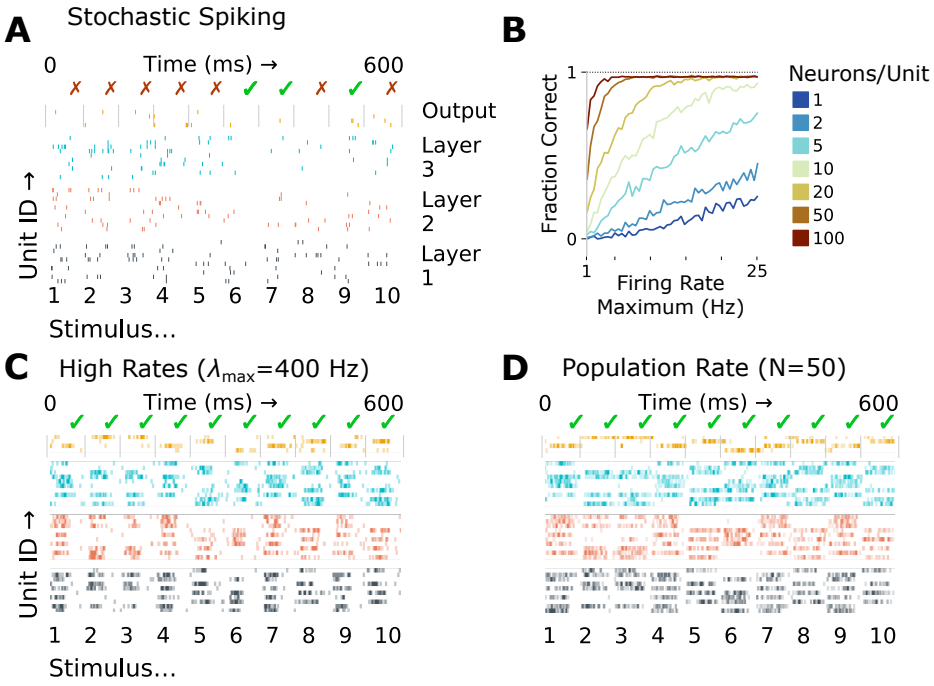


Fig. 2.3 A stochastic SNN based on ANN parameters does not work unless high firing rates or population rates are introduced. **(A)** Raster plot of the stochastic SNN attempting the classification task from Fig.2.2. For each stimulus (input point) on the x-axis, the tick or cross above the output denotes if the SNN classified it correctly or incorrectly. **(B)** Good classification performance can be achieved by increasing the maximal firing rate and/or by using a population code. **(C)** High firing rates restore the ANN performance. **(D)** Population rates can likewise restore performance. A population rate here was modelled as replacing an ANN unit with multiple copies of the SNN neuron, receiving / sending the same inputs / outputs to identical neurons.

of inputs spikes compensates for spike timing jitter. The problems with these solutions remain the same: the high firing rates required are unphysiological, and the population rate code is energetically costly, and amounts to having to use multiple unreliable neurons to do computations that could, in principle, be done by one unit.

Idealised LIF network

One can argue that the above is a straw man argument, as the SNN was set up to fail by building in unreliability into the neural dynamics, by making them inherently stochastic. So let us see what happens when we use deterministic spiking models. The simplest example of a spiking model is the LIF model, which is completely linear in its dynamics, and has a reset rule added on top: when the membrane potential reaches a threshold value ϑ, it gets reset:

$$\begin{aligned} \tau\dot{V} &= -V + I_{\text{syn}}(t) + I_{\text{app}}(t) \\ \tau\dot{s} &= -s \\ &\text{if } V \geq \vartheta : \\ &\qquad \text{set } V = 0 \\ &\qquad \text{set } s = s + 1. \end{aligned} \tag{2.11}$$

Here, I_{syn} is the synaptic current coming from the upstream neurons

$$I_{\text{syn}}(t) = \sum_{i \in \mathscr{I}}^{N} w_i s_i(t), \tag{2.12}$$

$\mathscr{I}$ is the set of all indices of the upstream neurons to which this neuron is connected, and $s_i(t)$ is the postsynaptic conductance of these neurons. I_{app} is an applied current that we choose; we set it again to be a constant current proportional to the analog inputs multiplied by the first weight matrix for the neurons in the first hidden layer, as before in the stochastic SNN, so that $I_{\text{app}}(t) = (W^1 \mathbf{p}_i^N) H(t - T_{\text{stim}}/4)$, and we set it to 0 otherwise.

What is the input-output function of such an LIF model? An LIF neuron with a threshold of $\vartheta = 1$ receiving spikes from one neuron with a weight of $w = 2.75$ spikes exactly once every time its upstream neuron spikes. That is, it fires spikes with the same frequency with which it receives them (Fig. 2.4).

This is convenient, because it easily allows us to train an ANN that has this same activation functions for its units: we use a ReLU function:

$$\text{ReLU}(x) = \max(0, x). \tag{2.13}$$

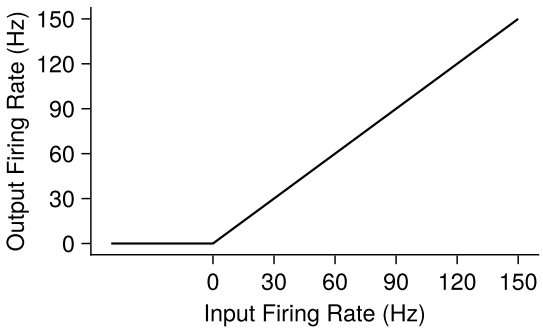


Fig. 2.4 The input-output function of a LIF neuron as described in Eqn. 2.11, which receives input spikes from one neuron with a weight of $w = 2.75$.

Then, if we ensure that the bias of every unit in the ANN is $b = -1$, and after training we scale all weight matrices by 2.75, the activation function of the ANN units matches the input-output rate function of the LIF neurons.

So we trained an ANN with the same architecture we had previously: entirely feedforward, with two input units, then three hidden layers with 10 units, and four output units. Again we trained on the toy problem from Figure 2.2, with the crossentropy loss function. The only difference is that the units are now equipped with ReLU activation functions, instead of the sigmoidal functions we used previously. The biases were initialised at -1 and were, as before, omitted from the set of trainable parameters. This ANN has a classification accuracy of 99.25%.

To simulate an SNN with the trained parameters, we multiplied the weights by 2.75 and simulated an LIF network with the same architecture as the ANN. To simulate the thus obtained SNN as closely to the rate-based ANN as possible, we computed the outputs ("firing rates") of the first hidden layer of the ANN, and triggered spikes in the first hidden layer with precisely those rates, by making the input neurons spike with an interspike interval (ISI) that is the inverse of the rate:

$$\mathrm{ISI}^i = t^i_{k+1} - t^i_k = \frac{1}{r^i} \quad \forall k,$$

where i denotes the ith neuron in the input layer, t^i_k its kth output spike, and r^i the output of the corresponding unit in the ReLU trained ANN, If we take the rate-interpretation of ANNs literally, this SNN should produce exactly the same results as the ANN.

We measured the accuracy as before, by counting the output neuron that spikes the most often during an interval as the answer given by the SNN. We also gave the SNN plenty of time to settle on an answer; we let it spike for a duration of 2 seconds. With this metric, the SNN performs at an accuracy of 56%; a reduction of accuracy of 43.25 percentage points

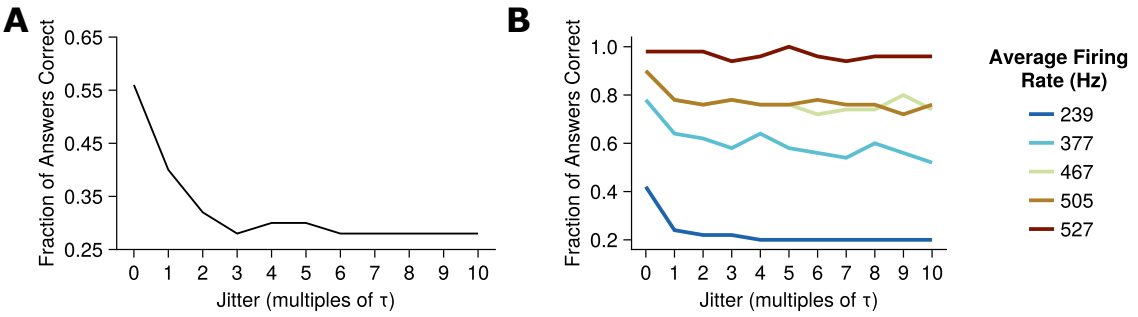


Fig. 2.5 **(A)** An SNN with LIF neurons, equipped with weight matrices obtained from training a rate-based ANN, does not perform well, and performance degrades rapidly in the presence of spike timing jitter. Jitter is a multiple of the neuron time constants τ. Accuracy of the original ANN was 99.25%. **(B)** Good ANN-like performance can be recovered, but requires unphysiologically high firing rates.

when compared to the original ANN. This shows that it is not entirely trivial to implement a rate-based ANN as an SNN.

Moreover, the LIF SNN performance drops off quickly to about chance levels when the input spikes to the network are jittered (Figure 2.5A). Here, the jitter was added by moving around the times of the input spikes to the network

$$t_k^{i,\text{new}} = t_k^i + \mathcal{N}(0,\sigma), \tag{2.14}$$

where t_k^i is again the kth output spike of the ith input neuron. As before, the standard deviation of the timing jitter σ is taken to be a multiple of the membrane time constant τ. It comes as no surprise that the situation can be remedied by simply increasing all the firing rates in the SNN, achieved by upscaling the weight matrices, but this has the drawbacks we already discussed.

It is worth pointing out that there exists a whole literature on converting ANNs to SNNs, in more sophisticated ways than we did in the above. However, the overall conclusion for these models remains the same: they do not use spikes parsimoniously, and they do not offer energy-efficiency benefits over regular ANNs [44]. More energy efficient solutions, as we have argued the brain should use, will consist of sparse spiking, and will therefore be more sensitive to noise in the spike timing.

This emphasises that any mechanism that might affect spike timing in any way, for instance by affecting the dynamics of the neurons, will have a significant adverse effect on the SNN performing its tasks. Therefore, mechanisms to deal with spike timing jitter are needed.

2.2 Single neuron mechanisms for reducing jitter

The results described in this section have been submitted as a manuscript [31] to a peer-reviewed journal. Some sections in this chapter were written as closely following to this manuscript. T. O'Leary and M. E. Rule contributed to the design of the experiments, but the computational results are all my own work.

We remarked in passing that the results we found in the above on how uncertainty in the timing and firing of output spikes quickly grows as timing jitter in input spikes increases hold for single-compartment neurons, not for neurons that have a spatial extent. In this section, we demonstrate that by adding compartments with specific properties to neurons, the difficulties of integration asynchronous input spikes can be remedied. These compartments are active dendrites, which are equipped with ion channels possessing slow dynamics.

So-called active dendrites are found in many species and in a great variety of neuron types; these are dendrites which produce a rich repertoire of nonlinear responses to synaptic inputs [84, 62, 26]. A class of well-studied behaviour of active dendrites are those found in cortical excitatory neurons [105, 137, 97], which provide long-lived positive feedback on the local membrane potential Ṫheir nonlinear behaviour fulfills many computational roles, and does not neatly fit into one category. The long-lived nature of these dendritic currents does suggest, though, that they will play some part in affecting the timing of action potentials in the neuron.

A conspicuous feature of cortical excitatory dendritic action currents is their duration of multiple tens of milliseconds [137, 136, 138, 106, 99, 97]. This stands out, because it is an order of magnitude longer than single synaptic inputs and axonal spikes (cf. Figure 2.6 middle and right). This extended duration of dendritic potentials in time seems at odds with rapid signalling and computation needs, and thus presents us with questions, particularly when cortical computations may involve relaying information over multiple brain areas in a short time interval [160]. Furthermore, the duration of dendritic events incurs a heavy energetic burden, because dendritic currents contribute significantly to the ATP budget of the brain [9]. What computational benefit could outweigh these signalling and metabolic costs?

We propose that the duration and threshold-like properties of dendritic currents can act as a mechanism embedded in the neurons to alleviate issues arising from spike timing jitter, as discussed in the previous section, thereby permitting robust computation in the face of spike timing variability. This is especially relevant to integration of inputs during high conductance states that are prevalent in-vivo. In these states the effective time-constant of the neuronal membrane is extremely short. Under such conditions, computations that require summation of multiple inputs place punishing constraints on spike timing precision. Figure 2.3 and Figure 2.5 show that the effect of spike timing jitter on network performance

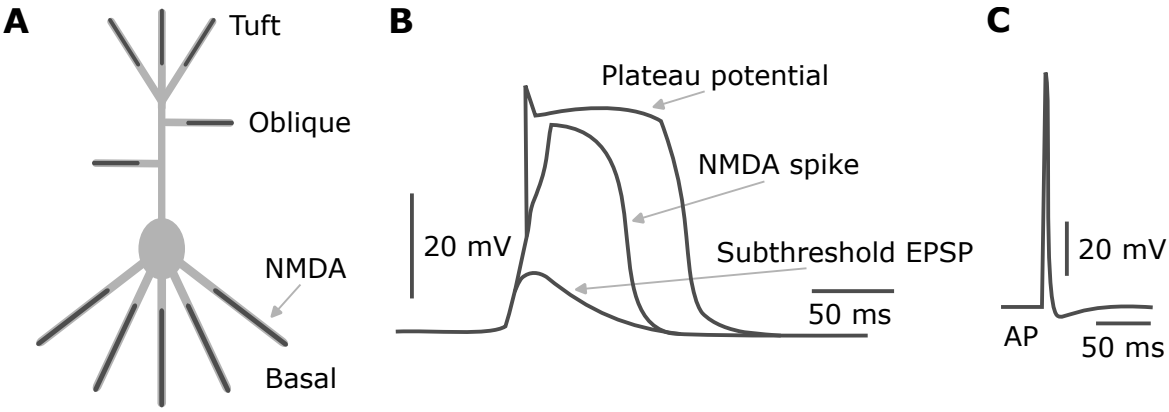


Fig. 2.6 Dendritic NMDA-dependent action currents. **(A)** Long lived calcium and voltage transients initiated within distal dendritic branches. **(B)** If the input to a dendritic branch with NMDA receptors is sufficiently strong, it can cross a threshold to produce an NMDA spike (bold trace). The NMDA response to inputs is super-linear, until it saturates in a plateau potential (top trace). **(C)** A somatic spike mediated by voltage-gated sodium channels. Note the order of magnitude difference in timescales with (B). Reproduced from [6].

must be seen as a function of multiples of the membrane time-constant; therefore, with very short membrane time-constants, these problems grow rapidly as jitter is introduced. Dendritic action potentials, by contrast, have a long duration that is dictated by the kinetic properties of voltage gated ion channels and NMDA receptors [136, 119, 29, 6], which can be predominantly found in distal dendrites of neurons (Figure 2.6A). These properties are largely determined by the amino acid sequence of receptor and channel proteins that are specifically expressed in dendrites [107, 106, 99]. This suggests dendritic properties are tuned to produce localised, suprathreshold events that outlive rapid fluctuations in the membrane potential. (cf. Figure 2.6B and C).

We extract core features of the complex biophysics of dendritic integration to construct and analyse a model, showing that rapid computation remains possible and is in fact facilitated by dendritic transients whose duration exceeds the integration time-constant of single neurons by an order of magnitude. We focus on computations that take place on the most rapid timescale because short integration windows are necessarily sensitive to timing jitter.

An interesting side product of this analysis is the interpretation of rapid cortical computations as binary network computations. The shortest window for a neuron to be still able to integrate inputs from multiple sources permits at most one spike being integrated from each source neuron. This notion of spiking networks operating in a binary regime is one we will return to in section 4.1. Here, we show how dendritic potentials in this regime allow for non-trivial, rapid spiking computations at the network level.

Numerous studies point out that threshold-like summation in dendrite can make neurons computationally equivalent to entire networks of simplified point models, or 'units' in a traditional neural network [25, 63, 97, 99, 107, 124, 125, 126]. The dynamical properties of dendritic action potentials further increase the range of computational possibilities by providing an additional long timescale over which the membrane potential can be influenced [64, 126, 24, 96, 125, 67].

The hypothesis that we are proposing is complementary to these proposed explanation on the purpose of regenarative dendritic action currents. With the dendritic potential as a backbone, the work we have done contributes to the computational toolkit of neurons by demonstrating how they can adjust their sensitivity to spike timing, thereby achieving reliable computations on fast timescales. Our work therefore puts forward the hypothesis that long-lived dendritic potentials somewhat paradoxically assist in the most rapid computations possible in a spiking network.

2.2.1 Abstract model of dendritic transients

Key features of NMDA action currents are their long duration and their super-linear integration of inputs [105]. Figure 2.7A and B show the recording of an NMDA spike from a cortical neuron in a rat, from [59]. [59] triggered an NMDA spike by glutamate uncaging, and measured the voltage response at the indicated (red, blue) sites. The voltage response (Figure 2.7B) shows that there is an orders-of-magnitude difference in timescale between an NMDA spike (left) and a sodium spike in the soma (right).

We took the biophysically and morphologically detailed computational model of [59], which represented a layer 5 pyramidal neuron [3]. It consists of 85 compartments and 439 segments, namely: 36 compartments for basal dendrites, 3 for the soma, 1 for the axon, and 45 for the apical dendrites. The model was implemented in NEURON (version 7.5) [73], and can be found on ModelDB (no. 249705). The channel parameters for different locations within the cell are given in Table 2.1.

In this geometrically extensive biophysical model, we simulated glutamate releases 50 ms apart in the three sites in the basal dendrites indicated in Figure 2.7C, a distance of 0.9 removed from the soma (where 1 corresponds to the length of the entire dendrite). The glutamate release was modelled as an instantaneous increase in the local NMDA conductance, thereby triggering three NMDA spikes at those sites. Despite these dendritic spikes being initiated 50 ms apart from one another, they are nevertheless summed together in the soma because of their long duration, and manage to trigger an action potential there (Figure 2.7D).

We created an abstract model of these NMDA spikes to capture the minimal requirements necessary to facilitate their role in circuit computations. This also allowed us to simplify the

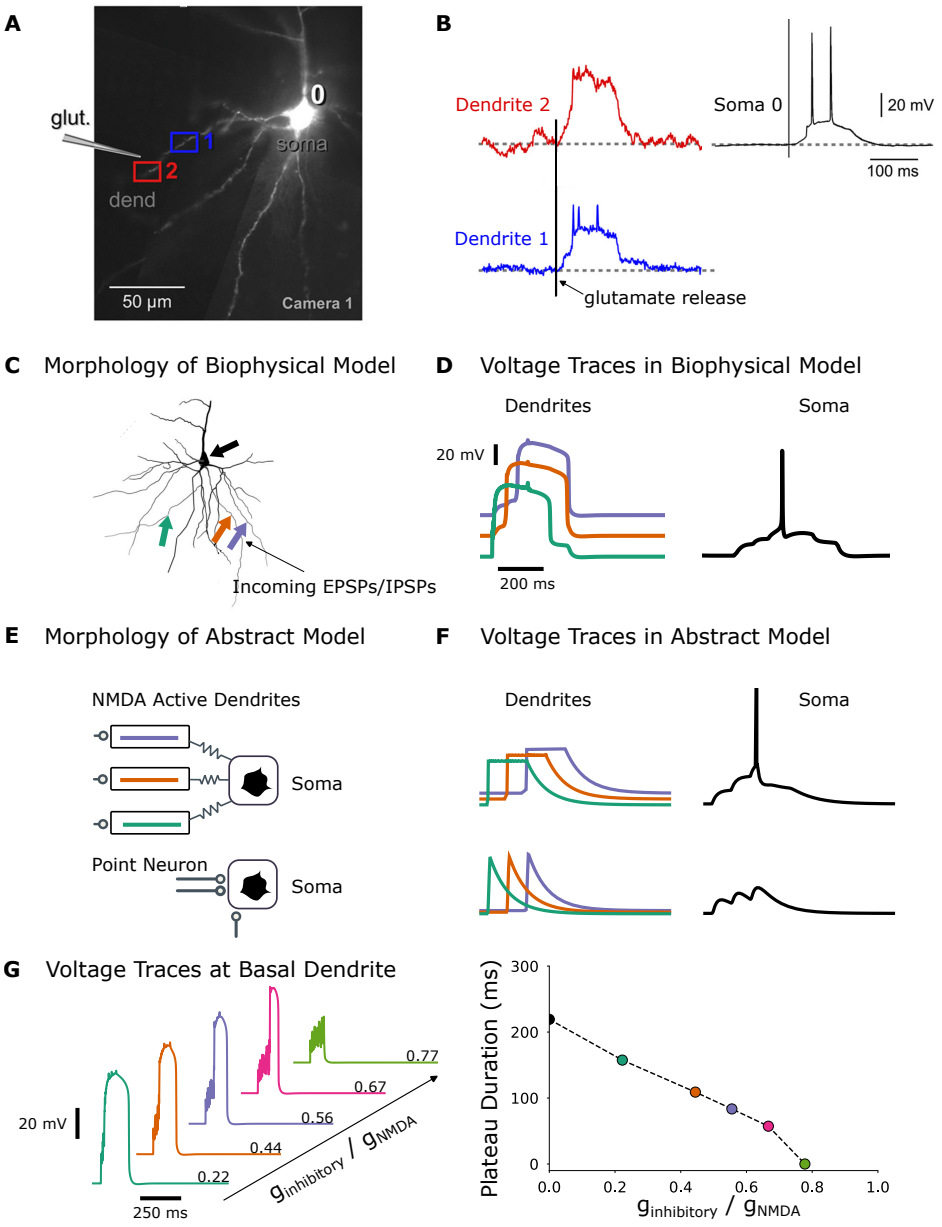


Fig. 2.7 **(A)** Photo of a cortical neuron in a slice of rat cortex from [59]. Two dendrites are marked where an NMDA spike is triggered through release of glutamate. **(B)** Voltage traces of the NMDA spikes of the dendritic patches marked in A. Note the long duration of the NMDA spike compared to the somatic sodium spikes evident in the top-left panel. **(C)** The morphology of the biophysical model used for simulating detailed NMDA plateau potentials. The arrows mark the positions where glutamate release was simulated. **(D)** The three traces of the NMDA spikes triggered at the sites marked in C, and the resulting somatic spike. **(E)** The morphology of the abstract model, with and without active NMDA dendrites. **(F)** The voltage traces of the abstract model, with and without plateaus. Because of the extended time duration of the plateau potentials, they sum accurately to produce a somatic spike. In the case where the plateau potentials are absent, they do not sum due to the short membrane time-constant of the soma. **(G)** Examples of NMDA plateaus in an active dendrite in the biophysical model as the inhibitory conductance $g_{\text{inhibitory}}$ is increased (left). The duration of the plateaus decreases approximately linearly as the inhibitory conductance increases (right).

Table 2.1 Specifications of the ion channel densities in the biophysical model of Fig. 2.7. 'Distance' refers to the location of the compartment / segment within their part of the neuron (soma / dendrite), measured from the soma, and ranges from 0 (at soma) to 1 (maximally removed from soma). Values taken from [59].

Channels	Conductance	Units
Voltage gated sodium channels		
Soma	900	
Basal dendrite	150 - $\frac{1}{2}$ distance	
Apical dendrite	375	pS/cm^2
Axon	j50	
Axon initial segment	1500	
A-type potassium channels		
Soma	150	
Basal dendrite		
Distal	$(150 + 0.7$ distance$)$	
Proximal	$(150 + 0.7$ distance$)$ $(1 - \frac{1}{300}$ distance$)$	$pS/\mu m^2$
Apical dendrite		
Distal	300 ($\frac{1}{300}$ distance)	
Proximal	300 ($1 - \frac{1}{300}$ distance)	
High-voltage activated calcium channels		
Soma	2	
Basal dendrite		
Distal	0.4	
Proximal	2	$pS/\mu m^2$
Apical dendrite		
Distal	0.4	
Proximal	2	
Low-voltage activated calcium channels		
Soma	2	
Basal dendrite		
Distal	1.6	
Proximal	2	$pS/\mu m^2$
Apical dendrite		
Distal	1.6	
Proximal	2	
HCN channels		
Soma	0.0001	
Basal dendrite	0.0001	S/cm^2
Apical dendrite	0.0002 (-0.8696 + 2.0.0870 exp(distance / 323))	
Calcium-activated potassium channels	1.68e-4	S/cm^2
K_v channel		
Soma	40	
Basal dendrite	40	$pS/\mu m^2$
Apical dendrite	40	
Axon	100	

model for computational efficiency. The model consists of a somatic compartment coupled passively to multiple dendritic compartments, each of which corresponds to a single branch on the dendritic tree (Figure 2.7E). The somatic compartment has standard LIF dynamics (see Equation 2.11). The dendritic compartments also have leaky dynamics, and we modelled the NMDA spikes by thresholding the dendritic voltage when it exceeds a threshold. When this threshold is reached, the dendritic voltage remains depolarized for some fixed time before returning to rest (Figure 2.7F).

In equation form:

$$\begin{aligned} \tau^d \dot{V}_i^d(t) &= -V + I_i^{\text{input}} - I_i^{sd} \\ I_i^{sd} &= g_i(V_i^d - V^s) \\ \tau^s \dot{V}^s(t) &= -V - I^r + \textstyle\sum_i I_i^{sd} \\ \tau^s \dot{I}_r(t) &= -I^r \\ \dot{s}(t) &= -s, \end{aligned} \tag{2.15}$$

here the superscripts d and s indicate variables belonging to the dendrite and soma respectively, I_i^{input} is the current triggered by a spike arriving at dendrite i, I_i^{sd} is the current flowing between dendrite i and the soma, with g the conductance between the two, $\tau^{d,s}$ is the time constant, I^r is a refractory current, and s is the postsynaptic conductance. When the dendrite reaches threshold Θ^d the dendrite remains there for P ms:

$$\text{if } V^d > \Theta^d : V_d = \Theta_d \text{ for } P \text{ ms}, \tag{2.16}$$

When the soma reaches its threshold Θ^s a spike is triggered:

$$\begin{aligned} &\text{if } V^s \geq \Theta^s : \\ &\qquad \text{set } V^s = 0 \\ &\qquad \text{set } I^r = I^r_{\max} \\ &\qquad \text{set } s = 1, \end{aligned} \tag{2.17}$$

i.e. the membrane potential is reset, a refractory current is activated, and the postsynaptic conductance increases. Note that the only effect of a spike on the dendritic dynamics is through I^{sd}, otherwise the dendritic dynamics are independent from the somatic dynamics.

The voltage dynamics are thus parametrised by the threshold and duration of the NMDA spike. We refer to this behavior as "Leaky Integrate-and-Hold" (LIH). This LIH model captures the key characteristics of the NMDA spikes, namely the threshold plus saturation of the super-linear integration, and the long-lived plateau of the dendritic voltage once the dendritic threshold is reached.

Due to the passive coupling between the dendritic and somatic compartments in the model, excitation in the dendrites depolarises the soma membrane potential, potentially leading to "axonal" output spikes. We do not model an axonal output compartment, we instead implement standard LIF dynamics in the somatic compartment, so that the spiking dynamics is given by Equation 2.17.

We compared the behaviour of our simplified model with that of the full, detailed biophysical model. To do this, we use the abstract model furnished with three active dendritic compartments. The dendrites were given input current pulses shaped as boxcar functions, with an amplitutde of 50 ms and a width of 1 ms:

$$I_i^{\text{input}}(t) = 50\left(H(t - t_{\text{input},i}) - H\left(t - (t_{\text{input},i} + 1)\right)\right), \tag{2.18}$$

$i \in \{1,2,3\}$. These input pulses were spaced 10 ms apart; $t_{\text{input},i} = 10i$ms. Each pulse is strong enough by itself to trigger the plateau in each dendrite.

The plateau potentials in the abstract model have a qualitatively similar effect on somatic membrane potential as the NMDA spikes in the biophysical model: Figure 2.7F, top, shows that spikes arriving at different times are still summed in the soma, and comparison with the biophysical model shows that the behaviour of the membrane potentials in dendrites and soma is qualitatively the same (cf. Figure 2.7F, top, and D).

We compared this to a situation where the same inputs arrive at a soma with standard LIF dynamics (Equation 2.11) and the same 1 ms membrane time constant. This time constant is consistent with the high-conductance state of pyramidal neurons in the cortex [18]. In contrast with the LIH model, inputs now decay after 2 ms to 3 ms, and thus fail to sum to spike threshold (Figure 2.7F, lower).

2.2.2 Dendrites transients aid integration of asynchronous spikes

As we just showed, the simplified LIH model captures a salient characteristic of the detailed biophysics of pyramidal neuron dendrites: the ability to integrate and hold inputs for a time which exceeds the membrane time constant. We hypothesized that this feature would be useful in situations where neurons need to integrate asynchronous input and reliably threshold it despite fluctuations in arrival times of the input.

If input spikes arrive in a narrow time window, reliably integrating them is trivial as we demonstrated previously (Figure 2.1). However, ensuring such synchronicity is extremely challenging in a large network that is subject to uncertainty and noise. This is easily illustrated in an artificial setting, and widely held to be true biologically; empirically, spike timing jitter is commonly observed at the population level [65, 153, 41]

We already looked at the severity of this problem by looking at how the spiking of a single-compartment Hodgkin-Huxley type model responds to timing uncertainty in its inputs (Figure 2.1). We repeat those experiments, now with our abstract model in hand, to demonstrate that the LIH model resolves some of the challenges posed by the requirement to reliably integrate asynchronous spikes. We will follow the methodology we used for the experiments in Figure 2.1. For one set of experiments, we used an LIF neuron, without active dendrites, and fed it input spikes. The times of these input spikes were drawn from a normal distribution. We took the standard deviation to be a multiple of the membrane time constant τ, which defines the timescale of the neuron dynamics, as before.

The results are consistent and virtually identical to the previous simulations done with the Hodgkin-Huxley type model: we see that the neuron has no trouble integrating spikes reliably, provided thay all arrive within a narrow time window (Figure 2.8A, B, left). But when the spikes arrive asynchronously, the neuron fails to integrate them reliably (Figure 2.8A, B, middle). Even small uncertainty in the input spike arrival times introduces noise in the membrane potential (Figure 2.8B, D), which can lead to the neuron failing to spike when it should have, or vice-versa. Asynchrony reduces the effective drive of the inputs (Figure 2.6C), which means that failure to spike will be more prevalent than erroneous spiking. This effective loss of excitation could be compensated by lowering the postsynaptic cell's threshold, but the output spiking variability due to the jitter still remains. This is shown in Figure 2.8D, where we use the coefficient of variation of the peak membrane potential to summarize the membrane-voltage uncertainty. Note that this uncertainty grows super-linearly as a function of timing jitter.

Having extended NMDA spikes remedies these issues (Figure 2.8A, B, right). Because of the extension of the duration of the spikes, the uncertainty of output firing is dramatically reduced, and the spikes are integrated as if they had arrived synchronously.

It is worth noting that, as in the case of the Hodkgin-Huxley model and the LIF model, these problems can also be addressed by increasing the number of inputs, thereby averaging away any uncertainty. Figure 2.8E shows that there is a linear relationship between the spike timing jitter in the inputs, and the number of input spikes necessary to have low uncertainty. Such an averaging procedure can only be done if timing variations are uncorrelated. This need not be the case, especially if timing jitter arises from variable conduction delays from common sources.

2.2.3 Active dendrites confer robustness in spiking computations

In the previous section we showed how a simple LIH mechanism, based on detailed biophysics, naturally extends the integration window for excitatory inputs. Although this concept

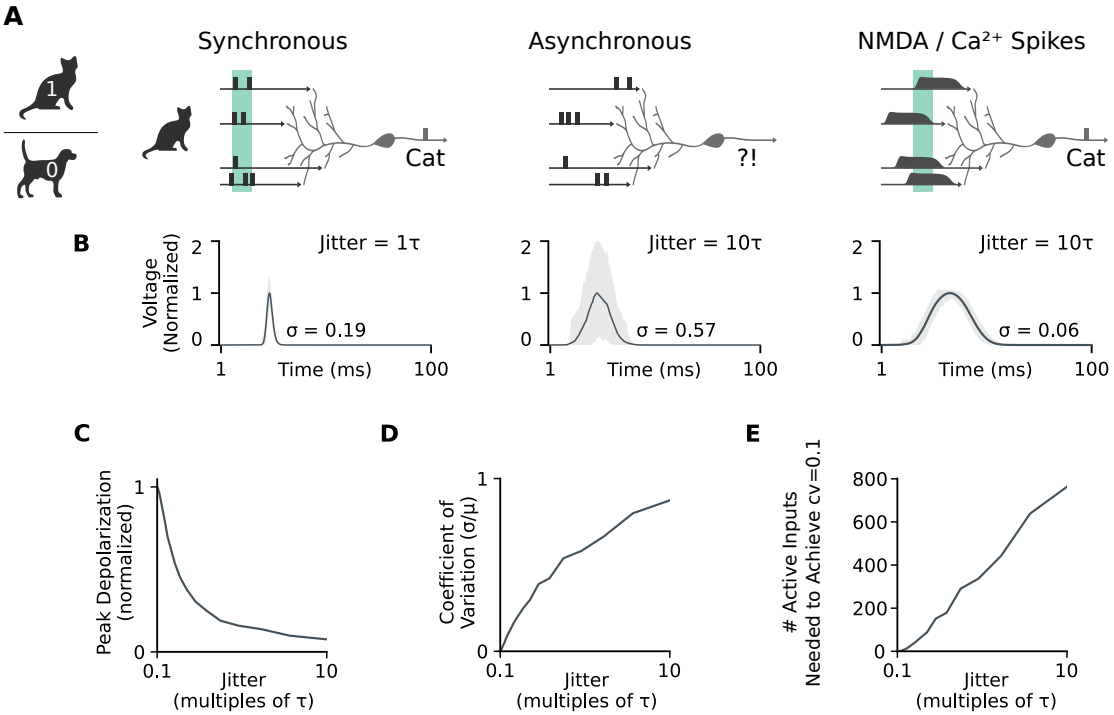


Fig. 2.8 **(A)** Neurons integrate inputs and compare the result to a firing threshold, which is comparable to performing a binary classification. When inputs are synchronous this can be done with a low number of spikes (left). But when spike timing is unpredictable (middle), this falls apart. Extended depolarizing potentials within dendritic compartments acts as a hold mechanism, allowing asynchronous events from different compartments to summate (right). **(B)** Simulation of summed voltage for 10 dendritic compartments, for small amounts of input-event timing jitter (on the order of one EPSP duration $\tau \sim 1$ ms; left), and larger amounts jitter (10τ, middle). Increased jitter increases the variability of the net depolarization. Extended depolarizing potentials on the duration of ~ 20 ms reduce the variability in net depolarization. **(C)** Increased jitter in the timing of input events reduces the net summed depolarization. **(D)** Increased jitter in the timing of input events increases the variability in membrane voltage depolarization. **(E)** Variability can be restored by increasing the number of inputs, but this is not cost-effective.

has potential, we still need to demonstrate how these dynamics can enable complex calculations in a network. How might we interpret the function of a neural circuit with NMDA spikes?

Input spikes arrive within some time window to elicit synaptic responses in a patch of dendrite, and the sum of these unitary synaptic responses either suffices to elicit a plateau potential or not, as shown in Figure 2.8. Consider the limit where the window in which spikes arrive becomes very small, for example because of time constraints on the computation. Such constraints do appear; see e.g. [154] for an example where neurons at each synaptic stage have about 10 ms to process presynaptic spikes and fire themselves, leaving room for the receiving and firing of about one action potential.

In this limit, each dendrite receives one spike, and itself either spikes or not: this can be interpreted as a binary computation, where incoming connections can be represented with a 1 (a spike arrives) or a 0 (absence of a spike), and the dendrite in turn produces a 1 (it fires) or a 0 (it does not). Connections between dendrites and soma are interpreted analogously: the dendrites produce 1s or 0s, and the soma sums these and compare the result to a firing threshold, thereby computing a 1 or 0. Neurons and dendrites operating in this regime have been observed, see e.g. [49, 164].

To show that excitable dendrites can support reliable computations, we use this binary interpretation. We built an SNN where the individual neurons consist of our abstract neuron model. We trained a BNN; a network where each unit's output is either 0 or 1, to perform a classification task (see: section 2.4.5 for details on the training). BNNs can be regarded as the saturated limit of regular sigmoidal networks, that is, a sigmoidal network with weights of large absolute value [102]. This interpretation is not restrictive: any binary function (function taking in and producing a vector of 1s and 0s) can in principle be done with a BNN [101, 146].

The task we train the BNN on is shown in Figure 2.9A. The 2D input points were first projected onto a binary feature space, to obtain vectors of 1s and 0s with 13 entries (Figure 2.9B). The dimensionality of 13 was chosen because this was the lowest dimensionality in which the binary network could still cope with the loss of information due to the binarization of the continuous coordinates. If the i^{th} input of the binary vector was a 1, a randomly generated timepoint t_i was added to produce an input spike (i, t_i), meaning that input neuron i was made to spike at time t_i. If the ith element of the binary vector was 0, it meant that neuron i would not fire for that input vector.

The network architecture is set up so that each dendrite received inputs from a unique upstream neuron. Naturally, each dendrite itself is only connected to one soma (see Figure 2.9C for a sketch). The assumption that each neuron connects to one dendrite of a downstream

neuron may seem strict, but is grounded in physiology: related inputs arrive at localised spike clusters on dendrites synchronously [152]. We have modelled these patches of dendrites that synchronously receive excitation from correlated inputs as one dendritic compartment.

After having a set of weights that gave good performance with the BNN we transplanted those weights to a spiking network with LIH neurons from section 2.4.3. This was done through setting the input currents to

$$I_i^{\text{input}} = w_{ij}^d s_j(t)\left(V_E - V_i^d\right),$$

where $s_j(t)$ is the postsynaptic conductance of neuron j, which projects on dendrites i with synaptic weight w_{ij}^d. Here, w_{ij}^d is the nonzero element of the ith row of the corresponding weight matrix in the BNN of the same architecture, trained with the so-called 'dendrite constraints' (see section 2.4.5 for details). This element is unique due to these constraints being enforced on the weight matrices during training.

This dendrite is coupled to its soma with weight w_{kl}^s to give the dendrite-soma current

$$w_{kl}^s\left(V_i^d - V^s\right),$$

with w_{kl}^s the (again unique) nonzero element of the kth row of the next BNN weight matrix, trained with the 'soma constraints' (section 2.4.5), and so on.

The 13 dimensional spikes were fed into the network by via the input layer. This layer consisted of 13 neurons: one for each element in the binary input vectors. As stated above, we generated a tuple (i,t_i) if the ith entry of the binary input vector equalled 1, with the t_i being drawn from a distribution. Then the ith neuron in the input layer would be made to fire at time t_i, by giving them the same boxcar-shaped input current as described in section 2.4.3.

We could then control the amount of (a)synchrony by varying the spike times t_i at which the input neurons spiked. Every input spike volley was centered around an input time t_{input}. Then the degree of asynchrony τ was controlled by drawing each spiketime, for each input neuron i, from a uniform distribution:

$$t_i \sim U(t_{\text{input}} - \tau/2, t_{\text{input}} + \tau/2).$$

To compare the performance of a spiking network without the dendritic plateaus to networks with plateaus, we removed the 'hold' on the dynamics of V_i^d for the former. Therefore, the dendrite potential would immediately start returining to its resting potential after reaching threshold. Otherwise, the circuit architecture remained unchanged. For Figure 2.9D, E the

input volleys were centered at $t_{\text{input}} = 15\,\text{ms}, 35\,\text{ms}, 55\,\text{ms}$, with an asynchrony measure of $\tau = 10\,\text{ms}$.

An answer given by the spiking network was considered correct if the ith neuron in the output layer spiked for an input point belonged to class i, and all other neurons in the output layer remained silent.

To compare the accuracies with and without plateaus in Figure 2.9F, we first generated spiketimes of the input neurons by drawing from the distribution for each data point and each value of τ, with the spike packet being centered at $t_{\text{input}} = 10\,\text{ms}$. Then these spiketimes were used for both network versions, and the accuracy of the network was measured as the percentage of points classified correctly by the network. In each simulation, each network saw one input vector only, to prevent interference from multiple overlapping inputs for high jitter values.

When the input neurons all spike exactly simultaneously, the spiking network mimics the BNN exactly, i.e. the same units produce outputs (either 1s or a spike) in both. But when asychrony is introduced, a discrepancy can arise. We examine a case of the SNN with and without receiving spikes in Figure 2.9D and E. In Figure 2.9D the dendrites are furnished with plateau potentials (as shown in Figure 2.7F, top). For these three asynchronous input spike volleys, the network gives the correct answer still. In fact, the identity of the neurons spiking still correspond exactly to those emit a 1 in the BNN counterpart.

This stands in contrast with the network performance without dendritic plateaus (Figure 2.9E), where the dendrites had no plateau potentials, as in Figure 2.7F, bottom. In Figure 2.9E it can be seen that the network now fails to process two of the three input vectors correctly. The duration of the dendritic spikes is too short so that the dendritic spikes are separated in time, and the soma fails to sum them all. To test how quickly this leads to a degradation of performance, we tested the accuracy (as percentage of inputs classified correctly) of the network as the asynchrony increased. The network with active dendrites coped well, but the performance of the network without dendrites with plateau potentials degraded rapidly (Figure 2.9F).

2.3 Discussion

An animal's ability to make quick decisions reliably is often crucial for its survival. This means that neural circuits will be under evolutionary pressure to be able to function at the fastest timescale possible. For instance, studies on primate visual reaction time suggest that each neuron in the synaptic pathway has a mere 10 ms to make a firing decision, allowing for only 1 to 2 spikes in each neuron on average [160, 154]. This puts the excitatory neurons

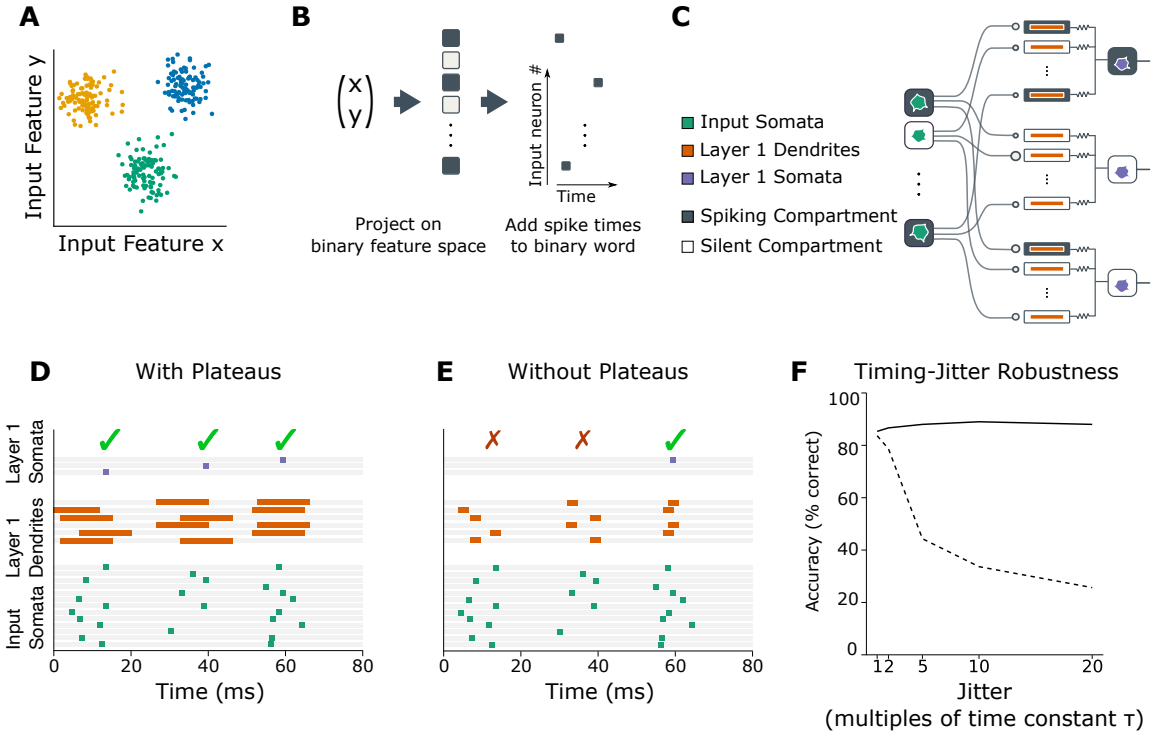


Fig. 2.9 A simple conductance based model displays the same qualitative behavior as a detailed biophysical model. **(A)** The classification task performed by the spiking network of figures D, E, F. Each point is a 2D input vector x, the colors represent the different classes. **(B)** Procedure of transforming continuous 2D inputs x into input spikes for the network. First, x is projected onto a binary feature space to obtain a binary vector in a higher dimensional space. Then, spiketimes are added to this binary vector to produce the series of input spikes. For details, see 2.4.5. **(C)** Schematic of the network architecture. The input somas spike according to the spiketimes obtained from the binary input vector. Each soma in the next layer has one dendrite per upstream soma, and each dendrite is connected to both one downstream and one upstream soma only. The dendrite-soma coupling is a bidirectional passive resistive coupling, whereas the upstream somas have a one-directional synaptic coupling onto the dendrites. **(D)** Example of the spiking network equipped with plateaus in the dendrites receiving asynchronous input spikes. It classifies the three inputs correctly in spite of the asynchrony. **(E)** Example of the spiking network equipped with dendrites without plateaus receiving asynchronous input spikes. The first two points are classified incorrectly, the network gets the third answer correct. **(F)** Summary of how well the network with plateaus and the network without plateaus deal with asynchrony τ, with the performance measured as the percentage of points of the classification task classified correctly. Without plateaus the performance drops off quickly, whereas the network with plateaus does not suffer from performance degradation for this range of τ.

in these pathways in an effective binary regime: either a neuron fires once during the entire computation, or it does not. Circuits in this regime will therefore not have high firing rates per computation (firing rates for a binary computation are not even defined). We examined how much of a problem this poses for neurons whose input spikes are subject to variability and uncertainty. We showed that, for low firing rates, the performance of naive SNN implementations is very fragile in the face of spike timing jitter. We then investigated how cortical neurons could leverage dendritic nonlinearities to make such rapid computations possible, even in biologically realistic scenarios involving spike timing jitter and signaling noise.

Dendrites, which were once believed to be passive cables, are now known to possess a variety of voltage-gated ion channels that generate nonlinear membrane potential dynamics. [148, 138, 97, 137, 86, 105]. These channels have been recorded in numerous types of neurons in the brain and are thought to have implications in neuronal communication, memory, and learning. As with axonal conduction, dendritic excitability provides a means for signals to overcome spatial attenuation. Therefore, it is not surprising to find regenerative currents in dendrites, particularly the long, thin dendrites of cortical neurons

It is far more puzzling why dendritic action currents are so much slower than their axonal counterparts. Their temporal dynamics, along with their nonlinear amplitude dependence opens endless ways for neurons to process time-varying signals. For instance, the dendrites of pyramidal neurons can perform complex tasks such as the discrimination of temporal signals [24] or the detection of coincident inputs [91]. In parallel with providing rich signal processing capabilities, dendritic currents also shape activity-dependent synaptic plasticity dynamics, and may thus allow neural circuits to learn temporal patterns [82, 74, 67].

We propose a complementary role for dendritic action currents that does not contradict any of these ideas, yet it addresses an outlying problem we believe is essential: how to make rapid cortical computation robust. Conduction delays and noise make asynchrony unavoidable in communication between circuits in the brain [53]. This poses a fundamental problem for the integration of related inputs: neurons with short membrane time constants, which are found in the high conductance regime in the cortex in-vivo, can only reliably integrate coincident inputs that arrive simultaneously within a few milliseconds of one another. Here, we have shown that slow time constants, which are provided by NMDA depolarization events within dendritic branches, can remedy the situation by widening the integration time window of neurons.

Our hypothesis is supported by several experimental findings. For example, research has shown that blocking NMDA receptors impairs precise spike-timing asynchrony and inter-area communication [174]. This hints at an important role for NMDA in facilitating reliable

synchronous communication between neuronal circuits. Recently it was shown that NMDA spikes are a major determinant of neuronal output in vivo, and that these dendritic spikes can be triggered by a handful of synaptic inputs [63, 106, 139, 25]. This is consistent with our hypothesis, which posits that NMDA spikes allow the network to perform computations with sparse spiking patterns.

An alternative solution that uses sparse spiking is for post-synaptic targets to become sensitive to specific, predictable, patterns of asynchronous spikes. Several computational studies have shown this is possible in principle using surrogate gradient learning rules that allow networks to perform computations based on relative spike-timings [117, 173]. However, these solutions are by design sensitive—rather than invariant—to the precise timing and order of inputs. It is therefore not clear that such solutions would work when networks are required to operate robustly on the fastest possible timescale.

Synchrony could potentially be maintained in networks that are organized as feed-forward "synfire chains", with relatively homogeneous transmission delays between nodes in each "rung" [1, 72]. [88] emphasize a role for refractoriness in maintaining synchrony, noting that post-spike inhibition "clips" late inputs, thereby maintaining a localized packet in time. [83] explore further the importance of dendritic nonlinearities in stabilizing packet synchrony.

The significance of these findings is to show that sparse yet reliable spiking computations may not require precisely synchronized inputs. This perceived necessity has sometimes been raised as an objection to the possibility of spiking computations based on spike times [98]. These objections are founded on the fact that function of neural networks can be affected by neuronal noise, as we have shown in this chapter, and many studies have tried to measure how much this can happen [15, 56, 176]. However, some of these studies may use a simplified model of a neuron that ignores its complex structure [60]. The presence of dendritic arbors, and their dynamic peculiarities, may reduce some of the negative effects of spiking noise.

Other, simpler solutions to the asynchrony problem have also been proposed. On possibility that we have mentioned at several instances in this chapter is that neural circuits exploit population averaging to overcome spike timing jitter [54, 142, 135, 90, 36]. However, this this is energetically costly [9] and would amount to scaling up the number of cells in a network to perform computations that could, in principle, be performed by more reliable single neurons. We also note that this idea only works when the spike timing variations are uncorrelated, which need not always be the case.

We, in contrast, propose that distal dendrites can perform efficient and reliable computation based on a threshold mechanism that uses NMDA currents, using fewer resources, with the tradeoff for neurons being that they have to generate and sustain plateau voltages. These potentials last longer and make the input timing less critical without additional learning,

reducing the number of neurons that need to spike to ensure reliable signal transmission. Our main prediction is that some circuits use these dendritic potentials to make coincidence detection more robust, allowing them to fire reliably even when the input timing varies on a larger timescale than the membrane time constant. We think these mechanisms are found in circuits that need to quickly identify or differentiate specific inputs, such as in the early stages of perception, or in circuits that coordinate long-range communication across different brain areas.

Table 2.2 Standard values of parameters used in the Hodgkin-Huxley model.

C	g_{Na}	g_K	g_l	E_{Na}	E_K	E_l
$1\,\mu\mathrm{F/cm^2}$	$120\,\mathrm{mS/cm^2}$	$36\,\mathrm{mS/cm^2}$	$0.3\,\mathrm{mS/cm^2}$	$50\,\mathrm{mV}$	$-77\,\mathrm{mV}$	$-55\,\mathrm{mV}$

2.4 Methods

2.4.1 Hodgkin-Huxley Neuron

The equations describing the Hodgkin-Huxley neuron are:

$$\begin{aligned}
\alpha_m(V) &= 0.1(V+40)/(1.-\exp(-0.1(V+40))) \\
\beta_m(V) &= 4\exp(-0.0556(V+65)) \\
\alpha_h(V) &= 0.07\exp(-0.05(V+65)) \\
\beta_h(V) &= 1/(1+\exp(-0.1(V+35))) \\
\alpha_n(V) &= 0.01(V+55)/(1.-\exp(-0.1(V+55))) \\
\beta_n(V) &= 0.125\exp(-0.0125(V+65))
\end{aligned} \tag{2.19}$$

$$\begin{aligned}
\infty(\alpha,\beta) &= \alpha/(\alpha+\beta) \\
\tau(\alpha,\beta) &= (\alpha+\beta)^{-1} \\
m_\infty(V) &= \infty(\alpha_m(V),\beta_m(V)) \\
n_\infty(V) &= \infty(\alpha_n(V),\beta_n(V)) \\
h_\infty(V) &= \infty(\alpha_h(V),\beta_h(V)) \\
\tau_m(V) &= \tau(\alpha_m(V),\beta_m(V)) \\
\tau_n(V) &= \tau(\alpha_n(V),\beta_n(V)) \\
\tau_h(V) &= \tau(\alpha_h(V),\beta_h(V))
\end{aligned} \tag{2.20}$$

The nominal values used to for our implementation of the Hodgkin-Huxley model are given in Table 2.2.

The equations describing the Hodgkin-Huxley neuron are:

$$\begin{aligned}
\alpha_m(V) &= 0.1(V+40)/(1.-\exp(-0.1(V+40))) \\
\beta_m(V) &= 4\exp(-0.0556(V+65)) \\
\alpha_h(V) &= 0.07\exp(-0.05(V+65)) \\
\beta_h(V) &= 1/(1+\exp(-0.1(V+35))) \\
\alpha_n(V) &= 0.01(V+55)/(1.-\exp(-0.1(V+55))) \\
\beta_n(V) &= 0.125\exp(-0.0125(V+65))
\end{aligned}$$

$$\begin{aligned}
\infty(\alpha,\beta) &= \alpha/(\alpha+\beta) \\
\tau(\alpha,\beta) &= (\alpha+\beta)^{-1} \\
m_\infty(V) &= \infty(\alpha_m(V),\beta_m(V)) \\
n_\infty(V) &= \infty(\alpha_n(V),\beta_n(V)) \\
h_\infty(V) &= \infty(\alpha_h(V),\beta_h(V)) \\
\tau_m(V) &= \tau(\alpha_m(V),\beta_m(V)) \\
\tau_n(V) &= \tau(\alpha_n(V),\beta_n(V)) \\
\tau_h(V) &= \tau(\alpha_h(V),\beta_h(V))
\end{aligned}$$

2.4.2 ANN classification

The classification task shown in Figure 2.2 consists of $N = 4$ classes, consisting of 100 points per class, with each class of points being defined by a Gaussian distribution. These Gaussian distributions are specified by the means

$$\mu_i^N = \{(2,5),(-6.3,2.1),(10,-7),(-4,-4)\}$$

and by the covariance matrices

$$\Sigma_i^N = \left\{ \begin{bmatrix} 2 & 0 \\ 0 & 2.5 \end{bmatrix}, \begin{bmatrix} 2.6 & 0.25 \\ 0.25 & 2.2 \end{bmatrix}, \begin{bmatrix} 2.6 & 0.25 \\ 0.25 & 2.2 \end{bmatrix}, \begin{bmatrix} 2.25 & 0 \\ 0 & 1.75 \end{bmatrix}, \begin{bmatrix} 2.4 & 0.05 \\ 0.05 & 2.25 \end{bmatrix} \right\}.$$

2.4.3 Leaky-Integrate and Hold Model

For all simulations using the LIH model (section 2.4.3), unless mentioned otherwise, the following parameter values that were used in simulations are given in Table 2.3.

Table 2.3 Standard values of parameters used in LIH model

τ^d	τ^s	g_i	Θ^d	Θ^s	$I^r_{\max}$	V_E
1ms	1ms	1 μS	1 mV	1 mV	5	100mV

2.4.4 Neuron with asynchronous inputs

For Figure 2.8 the abstract model was used. The spike times of the incoming spikes were drawn from a normal distribution. The jitter of the spike times was defined as the standard deviation of this distribution. This jitter should always be compared with the membrane time constant τ of the neuron, for which we used 1 ms.

We used jitter values of 1τ for the synchronous case and 10τ for the asynchronous case respectively, i.e. $t_{\text{spike}} \sim \mathcal{N}(0, \alpha\tau), \alpha \in \{1, 10\}$.

For all the computed values 500 simulations with a single neuron and randomly initialized spiketimes were performed. During each simulation, the neuron received 10 input spikes. For Figure 2.8 B, the solid line is the mean voltage over the ensemble of simulations, and the shaded region is the inner 90% of the voltage distribution.

In Figure 2.8 C,D,E all variables were computed inside the window of spikes arriving. The peak depolarization plotted is the mean of the ensemble of peak depolarizations for each simulation. Similarly, the coefficient of variation C_v plotted is the C_v for each jitter value, where that particular C_v is computed over all 500 simulations for that jitter value.

For 2.8 E the minimum number of input spikes was computed that would push C_v down to 0.1 for a particular value of the jitter value.

2.4.5 Binary network

The data consisted of 2D points which were assigned to different classes. The three classes were Gaussian clusters, with means $\mu_i = \{(4,5), (-12,2), (10,-7)\}$ respectively, and isotropic covariance of $\sigma = 1.5$. 100 points x_i per class i were generated from $x_i \sim \mathcal{N}(\mu_i, \sigma)$.

The task of the BNN was to classify the points correctly. Because we wanted to interpret the continuous 2D points as input spikes to our network, we binarized the data first. To this end the input vectors $x \in \mathbb{R}^2$ were transformed by mapping them onto $\{0,1\}^{13}$, i.e. every point was mapped onto a 13 dimensional vector of ones and zeros. This was achieved by randomly generating $W \sim \mathcal{N}_{13}(0, I)$ and $b \sim \mathcal{N}(0,1)$. Then we applied

$$x \mapsto \frac{1}{2}\left(\operatorname{sgn}(Wx + b) + 1\right).$$

To train our binary network we made use of surrogate gradients [117]. In short, we defined a function ϕ that for the forward pass (i.e. the network unit outputs) acted like a step function

$$\phi(x) = H(x).$$

But for the purposes of backpropagation the derivative of ϕ is defined to be

$$\phi'(x) = \frac{1}{4}\left(|\frac{x}{2}|+1\right)^{-2},$$

which is the superspike derivative [172], equivalent to the derivative of a fast sigmoid. Using this derivative allows us to train the network, in spite of the step function having a derivative that is zero almost everywhere.

The network weights were initialized with the distribution:

$$W \sim U(0,x),$$

where $x = \sqrt{6/(\text{no. inputs} + \text{no. outputs})}$.

The biases of the BNN units were set to -1, to enforce a positive firing threshold for the neurons. Note that this does not restrict the computational expressivity of the model compared to BNNs with trainable biases, as an ANN without trainable biases can be expressed as a smaller ANN with trainable biases. To see this, let us assume that we have a standard layer l in a feedforward ANN with trainable bias $b^l \in \mathbb{R}^m$:

$$\begin{aligned} z^l &= W^l x + b^l \\ a^l &= \sigma\left(z^l\right), \end{aligned} \tag{2.21}$$

where $x \in \mathbb{R}^n$ is the input to layer l, a^l is its output, and $W^l \in \mathbb{R}^{m\times n}$ its weight matrix. This is equivalent to:

$$\begin{aligned} z^l &= \left[W^l|b^l - c\right][x|1] + c \\ &= \sum_{i=1}^{n} \mathbf{w}_i x_i + b^l \\ &= W^l x + b^l \\ a^l &= \sigma\left(z^l\right), \end{aligned} \tag{2.22}$$

where $[x|y]$ denotes the horizontal concatenation of x and y, so that $\left[W^l|b^l\right] \in \mathbb{R}^{m\times(n+1)}$ and $[x|1] \in \mathbb{R}^{n+1}$, and $\mathbf{w}_i$ is the ith row of W. This shows that the larger network with c bias can

be made equivalent to a smaller network with trainable biases, and the argument extends easily without loss of generality to constant biases of any value.

Each soma was connected to one dendrite of each soma in the next layer. Let N_d be the number of dendrites per neuron for a layer, and N_o the number of somas. First an $N_d \times N_d$ diagonal matrix is constructed for each soma in the next layer. Then the total weight matrix will be the vertical concatenation of all these diagonal matrices.

Similarly, each dendrite is connected to only one soma. This is achieved by constructing an $N_o \times N_d$ matrix for output unit i where only the ith row is non-zero. The overall weight matrix is the horizontal concatenation of these matrices.

The weight matrices used by the network alternate between the matrices with 'dendrites constraints' (a matrix that projects onto a layer of dendrites), and matrices with 'soma constraints' (a matrix projecting onto a layer of somas).

For both weight matrices elements that are initially zero, remain zero during the training procedure, and the elements of the weight matrices were constrained to be nonnegative, by having the activation of a unit in the BNN be

$$\phi\left(|W|x+b\right).$$

We implemented this constraint because we modelled pyramidal neurons, which are excitatory neurons. Therefore, the weights from the connections of these neurons should be positive.

The BNN was trained using stochastic gradient descent and surrogate gradient methods. First the network outputs $\hat{y}(x)$ where put through the softmax function

$$s(\hat{\mathbf{y}})_i = \frac{e^{\hat{y}_i}}{\sum_{j=1}^{K} e^{\hat{y}_j}} \tag{2.23}$$

which was then parsed through the cross-entropy function

$$H(\hat{\mathbf{y}}(\mathbf{x}),\mathbf{y}) = -\frac{1}{N}\sum_{i=1}^{B}\sum_{j=1}^{K}\log(s_j(\hat{\mathbf{y}}_i))\,y_{i,j}, \tag{2.24}$$

where $\hat{\mathbf{y}}_i$ is the output in response to input $\mathbf{x}_i$ belonging to the batch of size B and $y_{i,j}$ the jth element of the corresponding target.

The stochastic gradient descent was performed by randomly shuffling the inputs and labels and choosing a batch size (30 in our case). Then in each epoch all batches were iterated over, and the parameters were updated using the Adam algorithm [87]. Training was terminated when the network gave the correct answer for more than 90% of all input points.

Chapter 3

Network mechanisms for reducing jitter

3.1 PING rhythms

In the previous chapter we showed that internal components in individual neurons can help make those neurons reliable. These internal components we discussed are active dendrites, and they can make neurons reliable in the face of timing jitter, which in turn helps confer robustness to spike timing uncertainty in SNNs.

In this chapter we will discuss how properties can emerge on a network level that can make neural circuits produce reliable spike responses, even when the individual neurons cannot do so independently. By 'reliable', here, we mean that the timings of action potentials in a population of neurons become synchronized, even when the inputs to, or individual parameters of, neurons vary. In other words, spike timing jitter is reduced. Specifically, we will discuss the role the so-called Pyramidal Interneuronal Gamma Rhythm (PING) plays in this. How these network properties could be exploited for computational purposes, we leave for chapter 4.

PING rhythms arise very naturally and easily in neural circuits with pools of excitatory and inhibitory neurons that feature a recurrent excitatory–inhibitory (E–I) connection, as shown in Figure 3.1. Such a configuration leads to *gamma* rhythms: an endogenous oscillation of neural firing in the 30 Hz to 100 Hz range. Though other mechanisms, such as inhibitory-inhibitory (I–I) models, also display gamma oscillations, the earliest model of such gamma oscillations was an E–I architecture [171, 162, 94, 20, 52, 38], and there is experimental evidence that such E–I driven gamma rhythms occur in the nervous system [22, 42, 155, 57].

To demonstrate how easy it is to obtain PING rhythms with an E–I model, we simulate pools of excitatory and inhibitory Hodgkin-Huxley type neurons (Equation 2.3). The internal dynamics of the neurons in both pools is the same, so their neuron type (i.e. excitatory or inhibitory) is defined by the outgoing connections from each neuron.

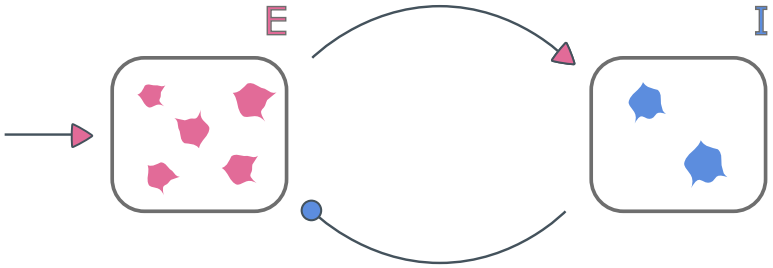


Fig. 3.1 The architecture of a neural circuit that can display PING rhythms. Pink and blue denotes excitatory (E) and inhibitory (I) neurons populations, respectively. Excitatory connections are shown as a pink arrow, inhibitory connections as a blue disk. The network is excited by external excitatory connections to the E population.

The network has a pool of N_E excitatory neurons, the synaptic connection of which to the I pool are mediated by AMPA receptors. The AMPA receptors are modelled as a current whose conductance is a function of the presynaptic membrane potentials:

$$\begin{aligned} I_{\text{AMPA}} &= g_{\text{AMPA}} \left(\frac{1}{N} \sum_{i \in \mathscr{I}}^{N} s_{\text{AMPA},i} \right) \left(E_{\text{AMPA}} - V_{\text{post}} \right) \\ \dot{s}_{\text{AMPA},i} &= \frac{1 + \tanh\left(V_{\text{pre},i}/10 \right)}{2} \frac{1 - s_{\text{AMPA},i}}{\tau_R} - \frac{s_{\text{AMPA},i}}{\tau_D}, \end{aligned} \tag{3.1}$$

where $\mathscr{I}$ denotes the set of the indices of presynaptic neurons connected to this neuron, so that $|\mathscr{I}| = N$ is the number of upstream neurons to which the neuron with membrane potential V_{post} is connected, and $s_{\text{AMPA},i}$ is the conductance as a function of the membrane potential $V_{\text{pre},i}$ of the ith upstream neuron. The equation for $\dot{s}$ is a standard model type for conductances; the s variable tracks the $1 + \tanh\left(V_{\text{pre},i}/10\right)/2$ term (which can be thought of as the normalized neurotransmitter concentration) with time constant τ_R and half-activation 0 mV, after which the conductance decays with time constant τ_D. The time-constants for AMPA have values of $\tau_R = 0.2$ and $\tau_D = 2$.

The inhibitory pool consists of N_I inhibitory connections, whose outgoing connections are modelled as being mediated by GABA_A, in the same fashion as the AMPA synapses:

$$\begin{aligned} I_{\text{GABA}_\text{A}} &= g_{GABA_\text{A}} \left(\frac{1}{N} \sum_{i \in \mathscr{I}}^{N} s_{\text{GABA}_\text{A},i} \right) \left(E_{\text{GABA}_\text{A}} - V_{\text{post}} \right) \\ \dot{s}_{\text{GABA}_\text{A},i} &= \frac{1 + \tanh\left(V_{\text{pre},i}/10 \right)}{2} \frac{1 - s_{\text{GABA}_\text{A},i}}{\tau_R} - \frac{s_{\text{GABA}_\text{A},i}}{\tau_D}, \end{aligned} \tag{3.2}$$

where all the symbols have the same meaning as before. The time-constant values are $\tau_R = 0.5$ and $\tau_D = 10$.

The reversal potential of $GABA_A$ E_{GABA_A} is low, so that I_{GABA_A} tends to decrease V_{post}, which is why it is an inhibitory current. E_{AMPA}, on the other hand, is high, so that I_{AMPA} increases V_{post}; therefore the current is excitatory. Note that another difference between the AMPA and $GABA_A$ dynamics are the time-scales: the AMPA time-constants are shorter (i.e. the AMPA dynamics are faster) than the $GABA_A$ time-constants. The equations for the ODEs of the synapse conductances are taken from [19]. For the values of the parameters used in the simulations, see section 3.4.1.

Using the model outlined above, we simulate the emergence of synchrony due to E–I mediated PING oscillations, the results of which are shown in Figure 3.2. We simulated 100 excitatory neurons and 25 inhibitory neurons. This 4:1 proportion is motivated by the experimental finding that in many areas of the cortex there are roughly four times as many excitatory as inhibitory neurons [23]. We define the connectivity probability $P_{E\to I}$ to be the probability of there being a synapse going from any neuron in the E population to any neuron in the I population, and $P_{I\to E}$ is defined analogously. Here, $P_{E\to I} = P_{I\to E} = 0.5$, so that, on average, an excitatory neuron will be sending to, and receiving from, half the inhibitory population, and vice-versa for an inhibitory neuron. We do not simulate any intra-population interactions (i.e. $P_{E\to E} = P_{I\to I} = 0$). The neurons in the E pool received constant applied currents randomly generated for each neuron $I_{app} \sim \mathscr{U}(10,14)$. We also added a $\pm 20\%$ variation in the internal maximal conductances of the neurons

$$g_i = g_i^0 X, \quad X \sim \mathscr{U}(0.8, 1.2),$$

where g_i^0 is the nominal value of the maximal conductances as given in Table 2.2, to ensure that synchrony could not arise accidentally due to the neuron models converging on the same limit cycle.

For the first half of the simulation, we set $g_{AMPA} = 0\,\mathrm{mS/cm^2}$, so that the neurons in the I pool received no inputs, thereby breaking the E–I loop (Figure 3.2A). Then, at $t = 250\,\mathrm{ms}$, we reinstated the loop by setting $g_{AMPA} = 1\,\mathrm{mS/cm^2}$. It is clear from both the raster plot (Figure 3.2B), which shows when each neuron spikes, and from the selected voltage traces from each of the neuron pools (Figure 3.2C), that initially the neurons in the I population are quiescent and the neurons in the E population are converging on limit cycles with varying frequencies. Then, when the E–I loop is restored, the neurons in the I pool become active, and both populations synchronize.

That the PING oscillation emerges so clearly, despite the randomly varying inputs to the E cells, the variability in the internal parameters of all neurons, and the random sparse connectivity pattern, is a strong indication of its robustness. In fact, the PING rhythm is also

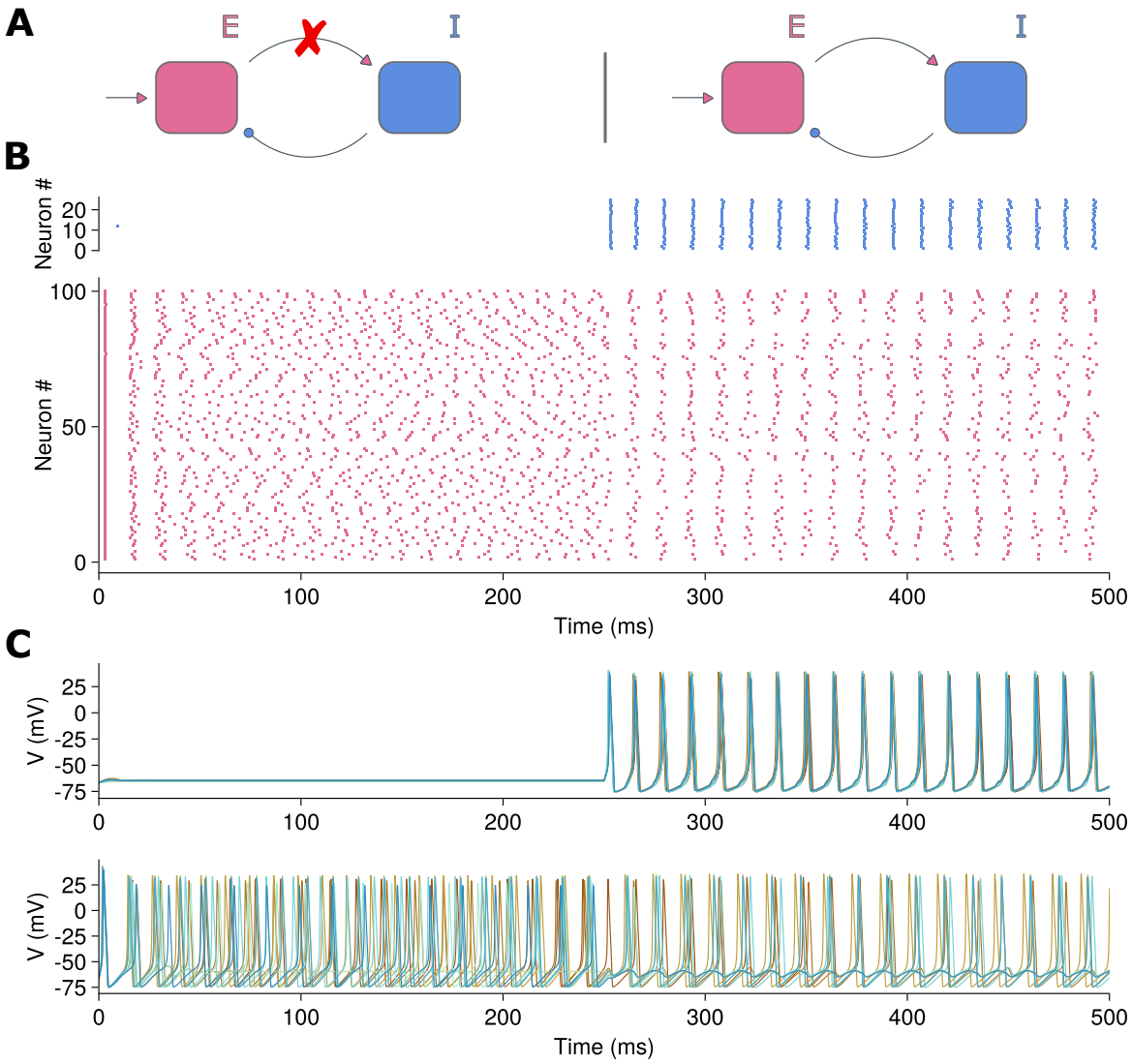


Fig. 3.2 An example of a PING rhythm due to E–I interactions with connectivity probabilities being $P_{E\to I} = P_{I\to E} = 0.5$ and $P_{E\to E} = P_{I\to I} = 0$. The E population consists of 100 neurons, the I population of 25 neurons. The neurons in the E population receive constant currents drawn from a uniform distribution, and internal maximal conductances are varied by $\pm 20\%$ for both neuron populations. **(A)** The E–I loop was broken (left) for the first half of the simulation, and then reinstated (right). **(B)** Raster plot of the spiketimes of the E (bottom) and I (top) pools of neurons. **(C)** Example voltage traces of 10 neurons from the I population (top) and the E population (bottom).

robust to the details of the internal dynamics of the neurons, and can be easily seen with an abstract neuron model like the theta neuron [20].

3.2 Origins of PING

The analysis above does not tell us where exactly PING oscillations originate from, except from some nebulous notion of an E–I feedback loop. We can exclude some possibilities, though:

1. PING does not depend on the network being in specific points in parameter space, as random parameter variability does not destroy the rhythm;
2. PING does not depend on specific E $\rightarrow$ I or I $\rightarrow$ E connectivity patterns, as random sparsity patterns sufficed;
3. PING does not rely on specific neuronal models, as different models (ranging from biophysically detailed to abstract) display the same rhythmic behaviour in the E–I setup.

A few conditions must be satisfied in order to have reliable PING rhythms, which were found heuristically [166, 156, 20]:

1. The E population must receive sufficient excitation so that a fraction of E cells spike that is large enough to excite the I pool (Figure 3.3A);
2. The E $\rightarrow$ I synapses must be strong enough so that a sufficient number of neurons in the I population receiving excitation reach threshold (Figure 3.3B);
3. The I $\rightarrow$ E synapses must be strong enough to synchronise the E population (Figure 3.3C);
4. Almost all the excitation in the I population must come from the E population, so that the I pool spikes in response to activity in the E pool (Figure 3.3D).

Several explanations have been given that have made these conditions more precise, for example through phase plane analysis of reduced models [20]. We have developed a new tool that gives a new perspective on the biophyics of the PING rhythm, and connects network-level excitation to single-neuron dynamics: the *population voltage clamp*.

The population voltage clamp extends the single-neuron voltage clamp technique as introduced by Hodgkin and Huxley [77]. Just as they used it to build a mean-field (i.e.

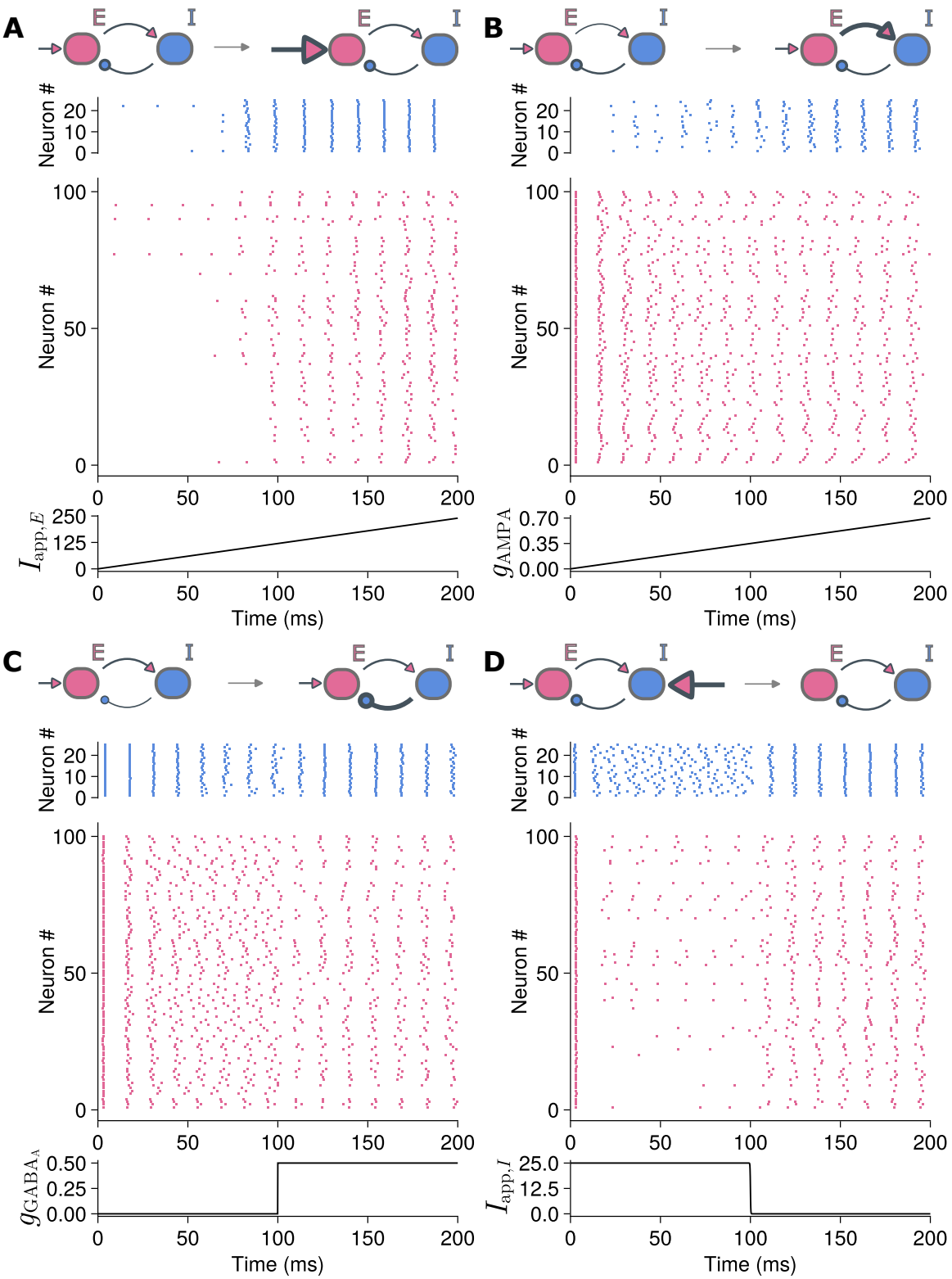


Fig. 3.3 Conditions for PING oscillations to arise in an E–I neural circuit. **(A)** A sufficiently large fraction of the cells in the E pool must be active. Here, we increased the excitation to the E population with time. **(B)** The E to I synaptic connections must be strong enough, so that the I population can get sufficiently excited by the E population. Here, we increased the value of g_{AMPA} with time. **(C)** The I to E synaptic connections must be strong enough, so that the I population can synchronize the E population. Here, we increased the value of $g_{\mathrm{GABA_A}}$ with time. **(D)** Most of the excitation in the I pool must be the result of spiking in the E population. Here, we gave each cell in the I population a driving current which decreased with time.

reduced) model of a single neuron, by lumping the thousands of ion channels in a neuron together in mean-field equations defined by the different ion types, the population voltage clamps lumps together the synaptic currents in a population of neurons. This is a similar approach as taken in Wilson-Cowan models [171, 170], which is the main mean-field approach for neural oscillations, but there are crucial differences. In Wilson-Cowan models, the spikes are not modelled. Instead, neural activity is assumed to be a continuous variable, which is usually summarised in separate excitatory and inhibitory variables, which are directly coupled to one another. In contrast, in the population voltage clamp, the spiking mechanism is still modelled (although the population is assumed to be perfectly synchronized), because the ion channels are included in the model, in contrast to Wilson-Cowan models. Instead, the synaptic interactions are modelled as ion channels as well, separated based on the type of synapse as in the Hodgkin Huxley model. This means that the mean-field models obtained through the population voltage clamp method remain more faithful to the biophysics of neurons than Wilson-Cowan models, as the conductances of the neurons are not excluded.

Including the conductances of ion channels into a mean-field model opens new avenues for scientific exploration. In the context of our spiking computation, we are interested in the reliability of PING for a set of parameter values such as maximal conductances and the weights of the synapses in the network. The mean-field model via the population voltage clamp offers a way of computing a criterion through the reliability of PING can be computed a priori for a network, which we explore in section 3.2.2. Being able to analyze which changes in parameter space change PING means we can study the modulation of the rhythm, and thereby the modulation of the computation that it facilitates. There are other potential benefits to having a mean-field model which includes conductances, of which we name a few. The parameter interpretability is better: conductances have clear biophysical meaning, namely ion channel densities, which makes it easier to relate model parameters to measurable quantities and incorporate experimental data into the model. It may allow us to make pharmacological predictions: by being able to model the effects of neurotransmitters or drugs that target specific ion channels, we can understand the impact of these substances on the neural oscillation. And we can understand multi-level integration better: because conductances provide a natural bridge between single neuron dynamics and large networks, they can be more easily integrated into models of vastly different spatial and temporal scales, from molecular dynamics to large-scale brain networks. Because in our framework PING facilitates the computation, all these benefits transfer immediately to the investigation of the spiking computation.

3.2.1 Computation of mean-field E-I current

In this section, we compute the mean-field E–I current. It consists mostly of a technical derivation of the equation and its parameters. The upshot is that we derive a mean-field current I_{EI} which qualitatively matches the experimental data. If one is not interested in this technical discussion, it can be safely skipped, and the reader can proceed to section 3.2.2.

The arguments from section 3.2 show that it is just the E–I interaction that is responsible for the synchronization of the neural populations. More precisely, it is the mean amount of interaction between the excitatory and inhibitory populations, that must be shared between a large enough portion within each population, that leads to the synchronization: each inhibitory neuron must receive excitation from a sufficiently large fraction of the E population, and vice versa for each excitatory neuron.

We can make a few more observations about these PING oscillations. In the mean-field limit, the 'main' variable of which all the other variables are a function is $\langle V_E \rangle$, the mean voltage of the excitatory neurons:

$$\langle V_E \rangle := \frac{1}{N_E} \sum_{i=1}^{N_E} V_{E,i}. \tag{3.3}$$

We call it the main variable because all other variables are connected to each other only via the effect they have on $\langle V_E \rangle$. This is analogous to V the Hodgkin-Huxley model (Equation 2.3); note that $\dot{m}, \dot{n}, \dot{h}$ depend only on m, n, h respectively, and on V, so not directly on each other. Therefore, they are coupled to one another via their influence on V. Hereafter, $\langle \cdot \rangle$ will denote an average over space, analogously to Equation 3.3.

All the (mean) gating variables of the excitatory neurons are obviously a function of $\langle V_E \rangle$, and the input to $\langle V_I \rangle$ is $\langle I_{\mathrm{AMPA}} \rangle$ which is itself a function of $\langle V_E \rangle$. Lastly, we observe that the synchronization within the E population is caused by $\langle I_{\mathrm{GABA_A}} \rangle$, which is, via $\langle V_I \rangle$, a function of $\langle V_E \rangle$. Therefore, all the relevant variables, in their mean-field limit, depend on $\langle V_E \rangle$, which is the variable which coupled all other variables indirectly to one another. In this sense, it is the main variable.

The idea behind the population voltage clamp is that the image sketched above is in some ways an analogy to single neuron dynamics: there is one main variable of which the other variables are a function; the membrane potential V in a single neuron (see Equation 2.3), and the mean membrane potential $\langle V_E \rangle$ in the E–I population. The system behaviour is caused by feedback loops of this variable on itself; the ionic currents in a single neuron, and additionally the synaptic currents in the population. But we know how to analyze and quantitatively describe these ionic currents; this is what Hodgkin and Huxley achieved in their seminal

work [75], using voltage clamps for single neurons. The population voltage clamp is to take this analogy at face value, and attempt to clamp the voltage of the E population to break the E–I loop and thereby analyze the synaptic currents in their mean-field limit (Figure 3.4).

We followed the same procedure that Hodgkin and Huxley invented to describe the three ionic currents of the squid giant axon [75, 77]. First, we clamp the voltages of the E population by applying a high gain, proportional negative feedback loop to the membrane potentials. We add

$$I_{\text{clamp}} = k\,(V_i - V_E) \tag{3.4}$$

to the $\dot{V}$ equation of every excitatory cell, where k is the gain and V_i is the voltage at which we clamp the neuron. The voltage clamp will go from $V_0 = -65\,\text{mV}$ to V_1, which we vary, thereby inducing a voltage step in the entire pool of E neurons (Figure 3.4A, middle). Hereafter, V_0 refers to the initial voltage of the population voltage clamp, and V_1 to the voltage after the clamp is used (see Figure 3.4A,middle).

Secondly, we start with the Ansatz that the net synaptic current on the E population, which is the result of the E–I loop, which we shall call $I_{EI}(t)$, can be described by first order dynamics:

$$\begin{aligned} I_{EI} &= g_{\text{GABA}_\text{A}} \underbrace{m^a h^b}_{\coloneqq g_{\text{pop}}} (V - E_{\text{GABA}_\text{A}}) \\ \dot{m} &= (m_\infty(V) - m)\,/\,\tau_m(V) \\ \dot{h} &= (h_\infty(V) - h)\,/\,\tau_h(V), \end{aligned} \tag{3.5}$$

where m and h are (in)activation variables. That is, we treat $I_{EI}(t)$ as the ionic currents are treated in Hodgkin-Huxley models (Equation 2.3). The dynamics equations of the (in)activation variables, $\dot{x}, x \in \{m,h\}$, as the result of a voltage step at $t = 0$ from V_0 to V_1 can then be integrated:

$$x(t) = x_\infty(V_1) + (x_\infty(V_0) - x_\infty(V_1)) \exp\left(\frac{-t}{\tau_n(V_1)}\right) \tag{3.6}$$

Then, in simulation, we recorded the synaptic current I_{EI}^{true}, where the 'true' indicates that this is the baseline data, which we seek to replicate. This quantity is defined as

$$I_{EI}^{\text{true}} = \langle I_{\text{GABA}_\text{A}} \rangle = \frac{g_{\text{GABA}_\text{A}}}{N_I} \sum_i s_{\text{GABA}_\text{A},i}\,(E_{\text{GABA}_\text{A}} - V_1)\,, \tag{3.7}$$

where we have taken into account that $V_{E,i} = V_1\,\forall i$ due to the voltage clamp on the E neurons.

We already know g_{GABA_A} and the electric driving force $(V_1 - E_{\text{GABA}_\text{A}})$, so we can divide I_{EI}^{true} by these values to find the time course of its conductance $g_{\text{pop}}(t) \coloneqq m^a(t)h^b(t)$. We can

find a and b by guessing some appropriate values, and seeing which values gives the best fitting curve for the total conductance.

For some voltage steps from a small enough $V_0 < -30\,\text{mV}$ to $-24\,\text{mV} < V_1 < -16\,\text{mV}$ we noticed that $h_\infty(V_1) \approx 0$ and $m_\infty(V_0) \approx 0$, by looking at the initial and steady-state values of g_{pop}, which are determined by $m_\infty(V_0)$ and $h_\infty(V_1)$ respectively. Furthermore, for low enough V_0, we can approximate $h_\infty(V_0) \approx 1$. Then we can use Equation 3.6 to find that for these voltage steps the conductance for a voltage clamp starting at t_{clamp} is given by

$$g_{\text{pop}} = m_\infty(V_1)^a \left(1 - e^{-(t-t_{\text{clamp}})/\tau_m(V_1)}\right)^a e^{-b(t-t_{\text{clamp}})/\tau_h(V_1)}. \tag{3.8}$$

When computing g_{pop}, we noticed that for intermediate values for V_1 there is a time delay before the conductance rises (Figure 3.4, left). This long time delay is well modelled by having a large a exponent. For aesthetic reasons, we set $a = N_I$. This makes the I_{EI} current match the picture of an ion channel, where a discrete gates have to open in order for the channel to be fully open, which gives the power relationship. In this case, the discrete subunits are the individual I neurons, of which there are of course precisely N_I, so setting $a = N_I$ is natural.

Using the approximation of Equation 3.8, we started with fitting $\tau_m(V_1)$ and $\tau_h(V_1)$. To do so, we optimised the τs so that for every V_1, our approximation would peak at the same time as the empirical conductance, and the decay time constant of the approximation and the empirical conductance matched.

Having then found $\tau_m(V_1)$ and $\tau_h(V_1)$, we were able to find $m_\infty(V_1)$ from Equation 3.8. This gave us enough datapoints to fit a sigmoid for $m_\infty(V)$; see section 3.4.3 for more details.

All that is left to do to specify the dynamics of our reduction, then, is to find h_∞. The inactivation h behaves differently from its Hodgkin-Huxley counterpart; at low voltages, the inactivation is always complete. This justifies setting $h_\infty(V) = 1 - m_\infty(V)$ for that voltage range. But at higher V values, it ceases to inactivate completely (Figure 3.4, right). This implies that h_∞ is not a monotonically decreasing function, which is perhaps questionable from a biophysical viewpoint.

We can proceed here in two different ways: first, we can find $h_\infty(V)$ at the higher voltage values by specifying it as $h_\infty(V) = (1 - m_\infty(V)) + h_{\text{rest}}(V)$, where $h_{\text{rest}}(V)$ is a sigmoid that saturates at around 0.3 (see section 3.4.3 for details). Or, we can simply set $h_{\text{rest}}(V) \approx 0$, as the τ_h is much greater than the timelength of a spike, and the voltage value spends little time in the range where h_{rest} would come into play. Indeed, both versions of h_∞ give the same qualitative behaviour for I_{EI}.

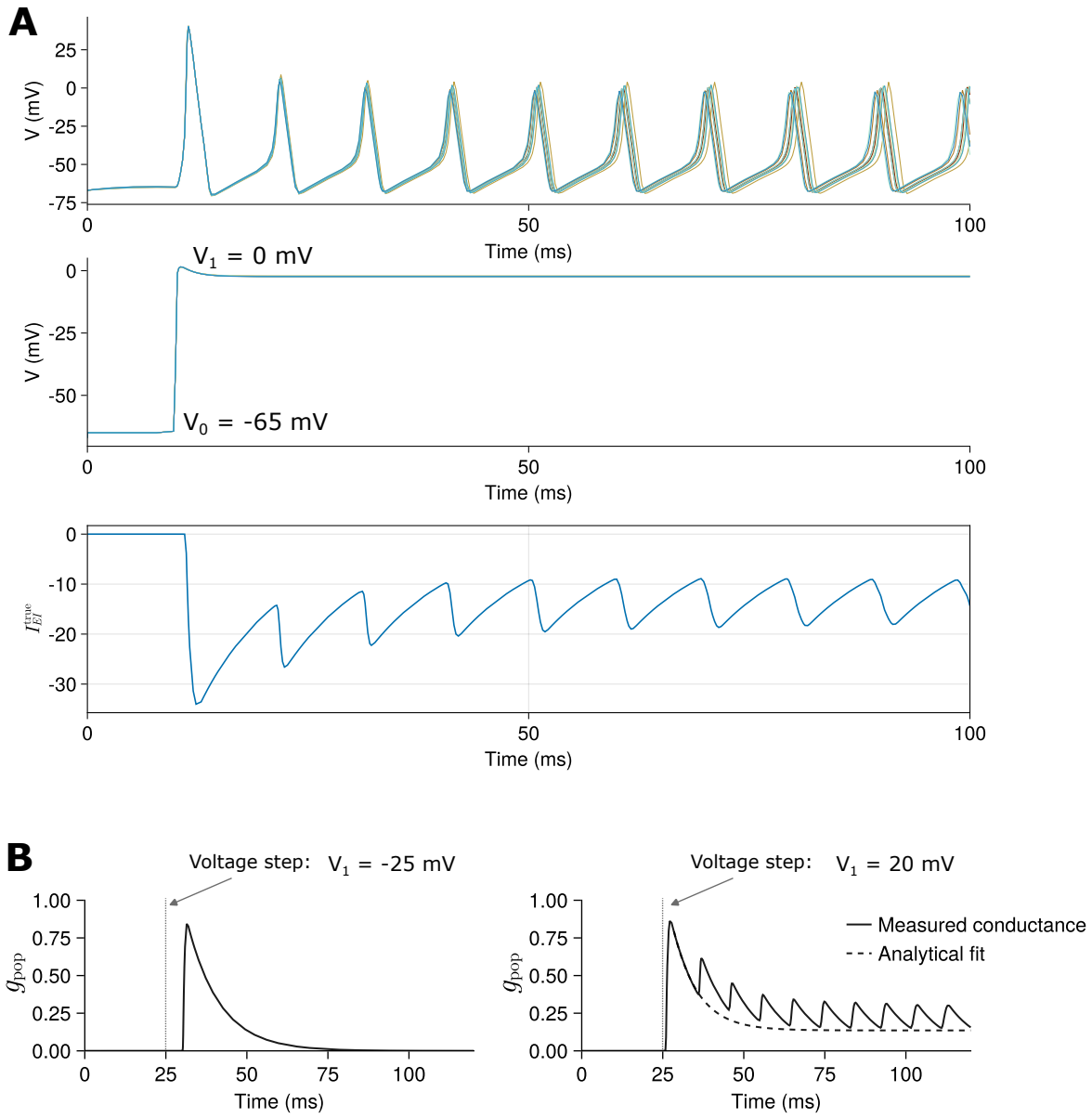


Fig. 3.4 **A** (top) 10 example voltage traces of I neurons, (middle) 10 example voltage traces of clamped E neurons, (bottom) the recorded I_{EI}^{true} current during the population voltage clamp. **B** Two timecourses for the dynamic part of the conductance $g_{\text{pop}} = m^{N_I} h$, for a voltage step from $V_0 = -65\,\text{mV}$ to V_1, one for $V_1 = -25\,\text{mV}$ (left), and one for a voltage step to $V_1 = 20\,\text{mV}$ (right). Both voltage steps were done at $t = 25\,\text{ms}$. Note that for the step to $-25\,\text{mV}$ there is a longer delay before the conductance rises and its inactivation is complete, whereas the conductance rises more quickly and does not completely inactivate for the step to the higher voltage value $20\,\text{mV}$. When the conductance displayed oscillatory behaviour, we worked with an analytical fit to its first peak, as shown by the dotted line in the right figure.

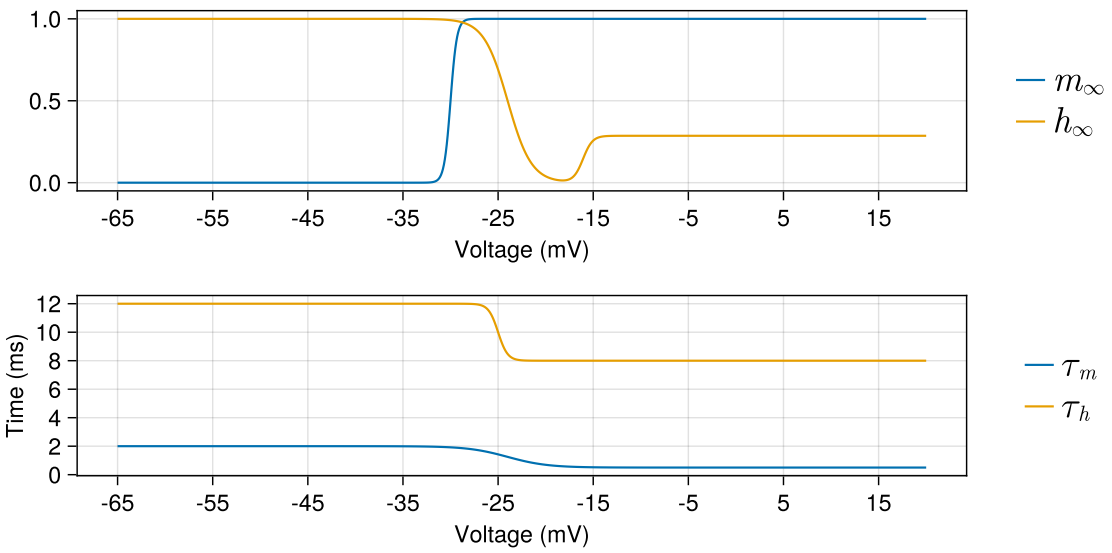


Fig. 3.5 The steady-state and time-constant functions of the m and h variables in the PING mean-field approximation.

The obtained steady-state values and the time-constant functions of m and h are shown in Figure 3.5. To show that these do indeed qualitatively give the correct behaviour, we first test the open-loop behaviour of I_{EI}. This means that we run the PING simulation with an E and an I population, and we let the E population excite our I_{EI} mean-field current, but we do not close the loop by connecting I_{EI} back to the E population. The outcome is demonstrated in Figure 3.6. We can see that qualitatively I_{EI} and the baseline I_{EI}^{true} match well, with the main difference being that I_{EI}^{true} does not decay to zero, but instead reaches a minimum value of around -4 nA. This is because I_{EI} is activated by $\langle V_E \rangle$, which activates m_∞ in a narrow time-window. In the regular PING setting, the I_{EI}^{true} is activated by the postsynaptic variables s coming from the E pool of neurons, where the s variables are functions of the voltage V_E. This means that the I population is, on average, activated as $\langle m_\infty(V_E) \rangle$, but $\langle m_\infty(V_E) \rangle \neq m_\infty(\langle V_E \rangle)$. In particular, $m_\infty(\langle V_E \rangle)$ is more sharply localized in time than $\langle m_\infty(V_E) \rangle$, which explains the discrepancy between I_{EI} and I_{EI}^{true} in Figure 3.6. One way of reconciling this discrepancy is by regarding the mean-field limit as the special case of a perfectly synchronized E population of neurons, where $V_{E,i}(t) = V_E(t) \, \forall i$ so that $m_\infty(\langle V_E \rangle) = m_\infty(V_E) = \langle m_\infty(V_E) \rangle$.

To test if this mean-field approximation actually works, we next test it in a closed loop setting (Figure 3.7). The picture of I_{EI} remains virtually unchanged compared to when it is receiving inputs in the open-loop PING network of Figure 3.6, and there is still clear PING-like rhythmicity in the firing times of the neurons in the E pool, as required.

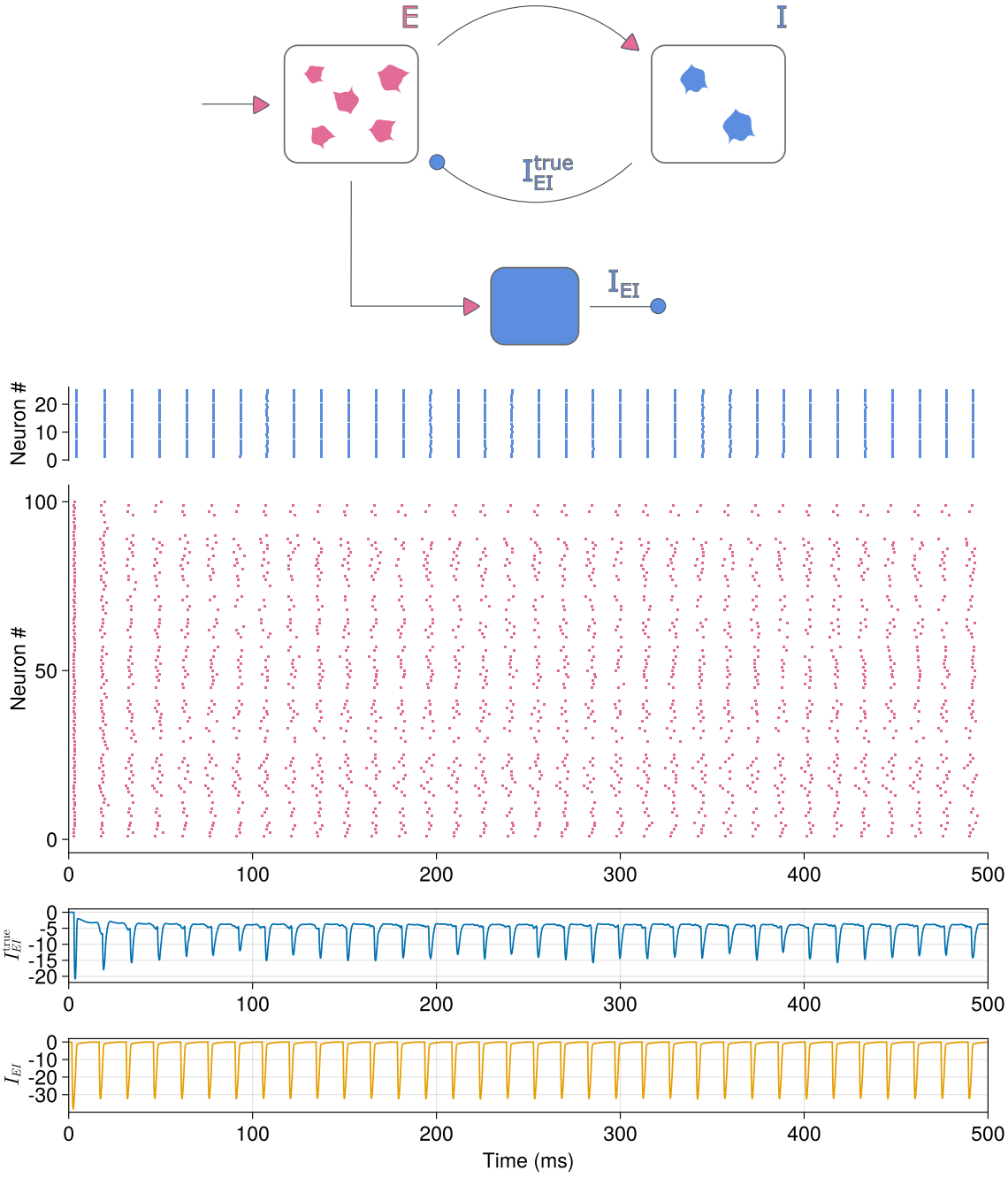


Fig. 3.6 Open loop PING behaviour in the mean-field approximation of the synaptic loop. Here, 'open loop' means that the I_{EI} mean-field approximation current is not fed back to the E population.

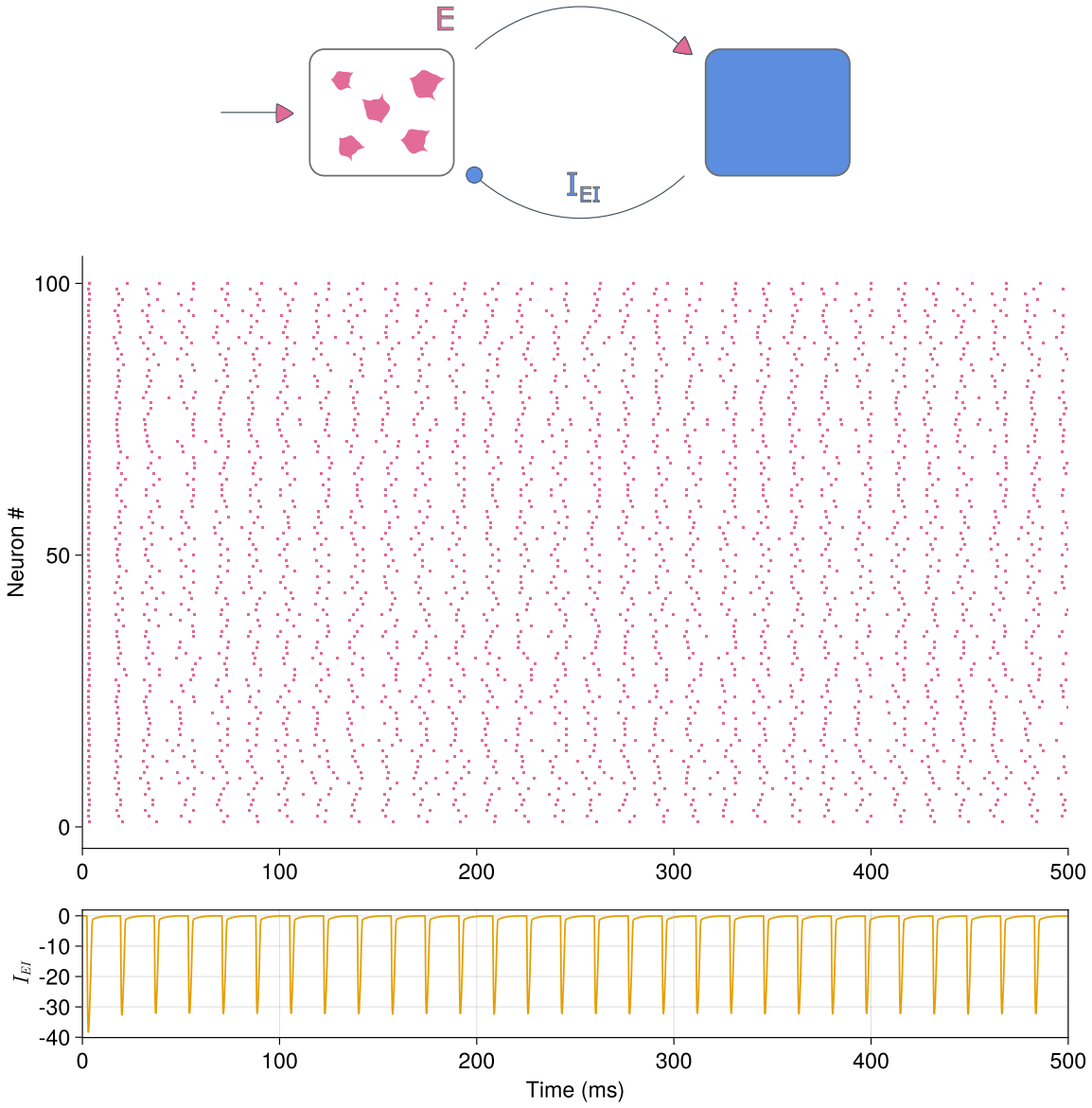


Fig. 3.7 Closed loop PING behaviour in the mean-field approximation of the synaptic loop. Here, the mean-field apprximation I_{EI} is fed back to the E population and also leads to synchronous PING-like behaviour.

3.2.2 Dynamic input conductance

What does this analysis tell us about PING rhythms? We can use previously developed methods [**drionDynamicInputConductances2015**, 55] to find the timescales in which the conductance of I_{EI} contributes to the neuron dynamics. Because the voltage dynamics are described by the equation

$$C\dot{V} = -\sum_i g_i(t,V)\,(V - E_i)\,,$$

for a sum over the i ion channels, a *negative* conductance $g_i(t,V)$ corresponds to *positive* feedback of that conductance on the membrane potential, because it will lead to a term of $\dot{V} \propto +V$, so that V will lead to in increase in itself. This kind of feedback separated in timescales is what leads to spiking. A single spike, for instance, is produced by fast positive feedback (e.g. provided by the Na current activation in the Hodgkin-Huxley formalism), followed up by slow negative feedback (provided by the K current activation and Na current inactivation in the Hodgkin-Huxley model) [**drionDynamicInputConductances2015**, 55].

Therefore, if we can dissect the conductance associated with I_{EI} in separate timescales, we can see in which timescales it gives feedback (positive or negative) on the membrane potential. We split up the conductance using the methodology given in [**drionDynamicInputConductances2015**, 55]. Let $\mathbf{w} = (w_1, w_2, \ldots, w_k)$ be the internal gating variables (m, n, h in the Hodgkin-Huxley model). Then

$$\begin{aligned} \dot{V} &= h(V,\mathbf{w}) \\ \dot{\mathbf{w}} &= f(v,\mathbf{w}) \end{aligned} \tag{3.9}$$

represents a general conductance-based model. In the Hodgkin-Huxley model 2.3 we have $h(V,\mathbf{w}) = -\sum_{ion} I_{ion}(v,\mathbf{w}) + I_{\text{app}}$ and $f(v,\mathbf{w}) = (f_1, f_2, f_3)$ where

$$f_j(V,w_j) = \tfrac{1}{\tau_j(V)}(w_{j,\infty}(V) - w_j)$$

captures the dynamics of the gating variables.

We can consider the transfer functions of the ionic currents in a linearized model neuron. (In the following $I_{\text{app}} = 0$.) Let I_i be the ionic current under investigation. We will consider this as a system that receives an input V and produces an output I_i. The system is given by

$$\begin{aligned} h(V,w_1,w_2) &= -g_i w_1^p w_2^q\,(V - E_i) \\ f_j(V,w) &= \tfrac{1}{\tau_j(V)}(w_{j,\infty}(V) - w_j), \quad j \in \{1,2\}. \end{aligned} \tag{3.10}$$

The output of this system is given by $y = h(V,w_1,w_2)$. Let V^* be an equilibrium voltage such that $h(V^*, w_1^*, w_2^*) = 0$ and $w_i^* = w_{i,\infty}(V^*)$ is the corresponding equilibrium gating variable

value. We linearize Equation 3.10 around the pseudo-equilibrium point (V^*, w_1^*, w_2^*) to obtain a new system

$$\begin{aligned} y_{\text{lin}} &= \left.\frac{\partial h}{\partial V}\right|_{(V^*,w_1^*,w_2^*)} V + \left.\frac{\partial h}{\partial w_1}\right|_{(V^*,w_1^*,w_2^*)} w_1 + \left.\frac{\partial h}{\partial w_2}\right|_{(V^*,w_1^*,w_2^*)} w_2 \\ \dot{w}_i &= \left.\frac{\partial f_i}{\partial V}\right|_{(V^*,w_1^*,w_2^*)} V + \left.\frac{\partial f_i}{\partial w_i}\right|_{(V^*,w_1^*,w_2^*)} w_i, \end{aligned} \tag{3.11}$$

where on the second line we made use of the fact that $\frac{\partial f_i}{\partial w_j} \neq 0 \iff i = j$.

We can evaluate these derivatives:

$$\begin{aligned} \left.\frac{\partial h}{\partial V}\right|_{(V^*,w_1^*,w_2^*)} &= -g_i w_1^{*p} w_2^{*q} \\ \left.\frac{\partial h}{\partial w_1}\right|_{(V^*,w_1^*,w_2^*)} &= -g_i p w_1^{*p-1} w_2^{*q} (V^* - E_i) \\ \left.\frac{\partial h}{\partial w_2}\right|_{(V^*,w_1^*,w_2^*)} &= -g_i w_1^{*p} q w_2^{*q-1} (V^* - E_i) \\ \left.\frac{\partial f_i}{\partial V}\right|_{(V^*,w_1^*,w_2^*)} &= w'_{j,\infty}(V^*)/\tau_j(V^*) \\ \left.\frac{\partial f_i}{\partial w_i}\right|_{(V^*,w_1^*,w_2^*)} &= -1/\tau_i(V^*), \end{aligned} \tag{3.12}$$

to find that

$$\begin{aligned} y_{\text{lin}} &= \left(-g_i w_1^{*p} w_2^{*q}\right) V + \left(-g_i p w_1^{*p-1} w_2^{*q} (V^* - E_i)\right) w_1 \\ &\quad + \left(-g_i w_1^{*p} q w_2^{*q-1} (V^* - E_i)\right) w_2 \\ \dot{w}_i &= \left(\frac{w'_{i,\infty}(V^*)}{\tau_i(V^*)}\right) V - \left(\frac{1}{\tau_i(V^*)}\right) w_i. \end{aligned} \tag{3.13}$$

To find the transfer function from V to w_i, denoted by $G_{w_i}(s)$, of this linearized system, we take the Laplace transform of the second equation in Equation 3.13:

$$G_{w_i}(s, V^*) := \frac{W_i(s)}{V(s)} = \frac{w'_{i,\infty}(V^*)}{\tau_i(V^*)s + 1}, \tag{3.14}$$

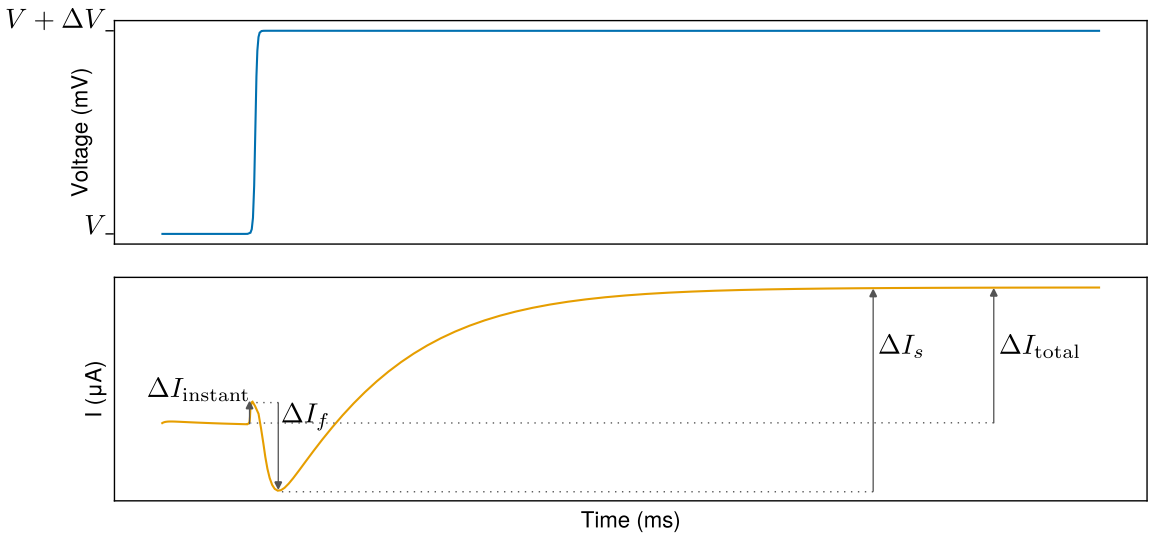


Fig. 3.8 Change in current in a Hodgkin-Huxley model during a voltage clamp. The total current ΔI_{total} induced by ion channels for a neuron undergoing a voltage clamp from V to $V + \Delta V$ can be decomposed in an instantaneous ($\Delta I_{\text{instant}}$), a fast ($\Delta I_f$), and a slow ($\Delta I_s$) component.

where $W_i(s)$ and $V(s)$ denote the Laplace transforms of w_i and V. We use this in the first line of Equation 3.13 to find the transfer function from V to Y which we denote by $G(s)$

$$\begin{aligned} Y(s,V^*) &= \left(-g_i w_1^{*p} w_2^{*q}\right) V(s) + \left(-g_i p w_1^{*p-1} w_2^{*q} (V^* - E_i)\right) W_1(s) \\ &\quad + \left(-g_i w_1^{*p} q w_2^{*q-1} (V^* - E_i)\right) W_2(s) \\ \therefore G(s,V^*) \coloneqq \frac{Y(s,V^*)}{V(s)} &= \left(-g_i w_1^{*p} w_2^{*q}\right) + \left(-g_i p w_1^{*p-1} w_2^{*q} (V^* - E_i)\right) G_{w_1}(s) \\ &\quad + \left(-g_i w_1^{*p} q w_2^{*q-1} (V^* - E_i)\right) G_{w_2}(s). \end{aligned} \tag{3.15}$$

We would like to evaluate in which timescales the transfer function of this current is active. Any conductance-based neuron model can be approximated with arbitrary precision by any parallel series of causal first order filters, plus a nonlinearity [35, 120, 134, 21]. Moreover, to capture all qualitative contributions of the ion channels, a lower number of timescales is needed: one faster positive feedback loop and one slower negative feedback loop per threshold for the excitable system [**drionDynamicInputConductances2015**, 55, 128, 158]. For example, the Hodgkin-Huxley model is an excitable system with one threshold, namely the threshold between quiescence and spiking. It turns out that this behaviour is fully captured with fast positive feedback and slow negative feedback on the membrane potential (Figure 3.8).

We can use this method to describe an excitable system with two thresholds, of which a neuron capable of bursting is a prominent example. The extra threshold lies below the fast threshold, and acts on a slower timescale, and is the switch that pushes the neuron away from rest, towards the fast threshold. In this way the slow switch leads to a sequence of rapid switching, before returning to rest.

We find the contribution of the transfer function Equation 3.15 around V^* to the feedback loops in the different timescales by using the decomposition

$$G(s,V^*) = G_0(V^*) + G_f(s,V^*) + G_s(s,V^*) + G_u(s,V^*), \tag{3.16}$$

where we evaluate a feedback loop in a *fast* (G_f), *slow* (G_s), and *ultraslow* (G_u) timescale. In principle, we could evaluate the effect of ionic currents in more timescales, but in practice neurons are very well described by dynamics in at most six timescale separations [128, 158, 159].

In particular, by evaluating this transfer function at $s = 0$, we recover the static input conductance which are given by differentiating $h(V, w_\infty) = I_{i,\infty}(V)$:

$$G(0,V^*) = \left.\frac{dI_{i,\infty}}{dV}\right|_{V^*} \tag{3.17}$$

Evaluating the transfer function at zero frequency gives the ratio of the steady-state output in response to a step input, in our case $\frac{\Delta I_{i,\infty}}{\Delta V}$. This is exactly what we measure whenever we do voltage clamp experiments with voltage steps of small amplitude ΔV. Therefore, by evaluating the transfer functions in different timescales G_α we can trace the dynamical evolution of the membrane current in response to a voltage step.

The time constants τ_{w_i} determine to which timescale the ionic current contributes. Three characteristic timescales were mentioned in the above, namely the fast, slow, and ultraslow timescales. We take the fast timescale τ_f to be the timescale of the fastest depolarizing current, which is sodium, therefore $\tau_f(V) = \tau_{m_{\mathrm{Na}}}(V)$. The slow timescale is defined as the timescale of the fastest repolarizing current, which in our case is potassium: $\tau_s(V) = \tau_{m_{\mathrm{K}}}(V)$. And the ultraslow timescale we take to be the slow timescale plus a constant: $\tau_u(V) = \tau_s(V) + 50\,\mathrm{ms}$.

The remaining gating variables are given a weight W^α that determines which timescale they contribute. The weights are defined as

$$\begin{aligned}
\mu^f(V;w_i) &= \gamma_{fs}, \\
\mu^s(V;w_i) &= \gamma_{su}(V;w_i) - \gamma_{fs}(V;w_i), \\
\mu^u(V;w_i) &= 1 - \gamma_{su}(V;w_i),
\end{aligned} \tag{3.18}$$

where the γs are given by

$$\gamma_{fs}(V;w_i) = \begin{cases} 1 \text{ if } \tau_{w_i}(V) \leq \tau_f(V) \\ \log\left(\frac{\tau_s(V)}{\tau_{w_i}(V)}\right)\log\left(\frac{\tau_s(V)}{\tau_f(V)}\right)^{-1} \text{ if } \tau_f(V) < \tau_{w_i}(V) \leq \tau_s(V) \\ 0 \text{ if } \tau_s(V) < \tau_{w_i}(V) \leq \tau_u(V) \\ 0 \text{ if } \tau_u(V) < \tau_{w_i}(V), \end{cases} \tag{3.19}$$

$$\gamma_{su}(V;w_i) = \begin{cases} 1 \text{ if } \tau_{w_i}(V) \leq \tau_f(V) \\ 1 \text{ if } \tau_f(V) < \tau_{w_i}(V) \leq \tau_s(V) \\ \log\left(\frac{\tau_u(V)}{\tau_{w_i}(V)}\right)\log\left(\frac{\tau_u(V)}{\tau_s(V)}\right)^{-1} \text{ if } \tau_f(V) < \tau_{w_i}(V) \leq \tau_u(V) \\ 0 \text{ if } \tau_u(V) < \tau_{w_i}(V), \end{cases}$$

i.e., we measure the relative distance between the timescales and normalise by the width of the gap between the timescales.

Putting all this together, the contribution of current I_i to timescale $\alpha \in f,s,u$ is given by

$$G_\alpha(V^*) = \sum_{w_i} \mu^\alpha(V^*;w_i)\, \frac{dI_{i,\infty}}{dw_{i,\infty}} \frac{dw_{i,\infty}}{dV}\bigg|_{V^*} \tag{3.20}$$

where we sum over all the gating variables w_i belonging to current I_i.

Using this method on our mean-field approximating I_{EI} of the PING mechanism (Equation 3.5), we found the conductances as shown in Figure 3.9. As one would expect for an inhibitory current, there is mostly negative feedback in the fast timescale, as the conductance there is mostly positive (Figure 3.9, top). The slower conductances are more revealing: in the slow timescale, there is a mixture present of positive and negative feedback (negative and positive conductances) (Figure 3.9, middle). Similarly, the conductance in the ultra-slow timescale is entirely negative, so it provides positive feedback (Figure 3.9, bottom). This positive feedback in the slower timescales is interesting, because slow positive feedback has been connected to the robustness, modulation, and tunability of circuit rhythms [**franciRobustTunableBursting2017**, 48], an idea which we will revisit later.

The slow positive feedback of PING resides in the internal dynamicso of the neurons in the I pool. More precisely, the auto-inhibition of the I population provides slow positive feedback on the E neurons, analogous to an inactivation of an outward current [**drionDynamicInputConductances201**
This self-inhibition can be the result of the intra-neuron dynamics, which cause a refractory period, or from I–I inhibition (which we have ignored so far).

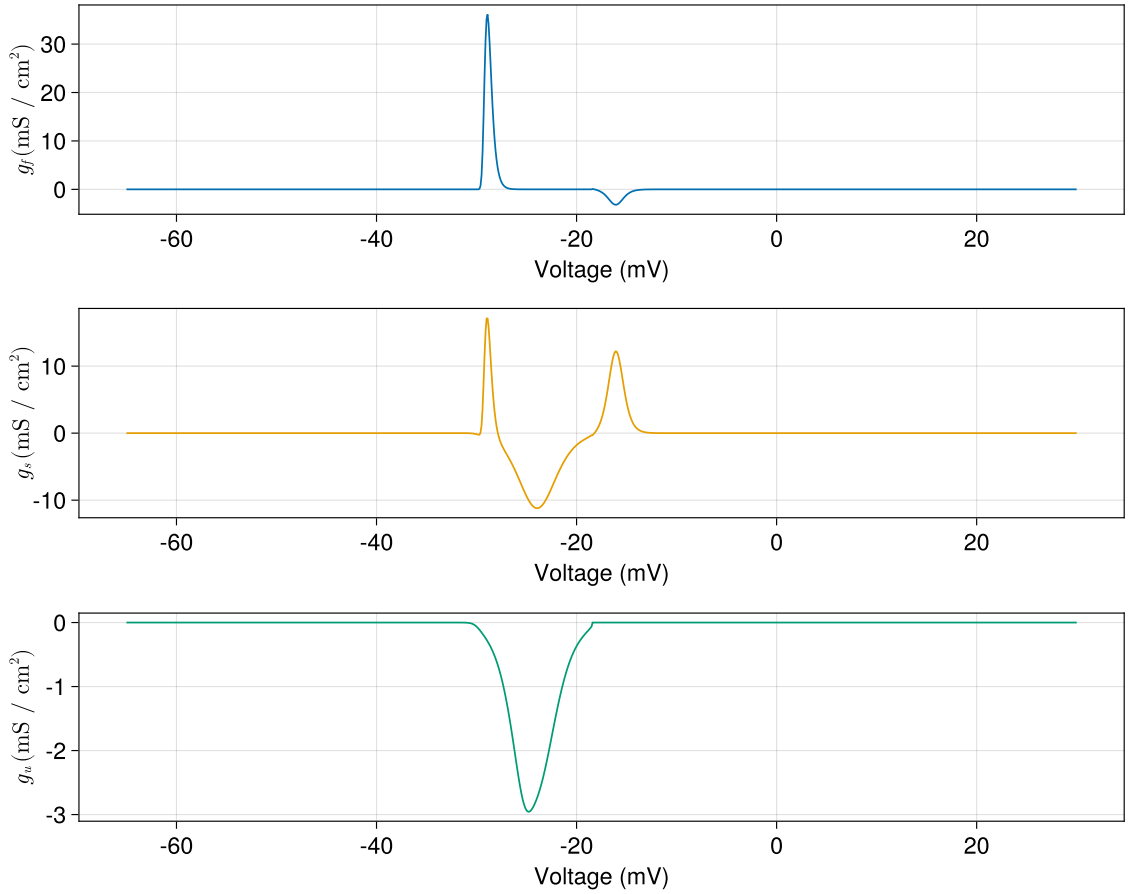


Fig. 3.9 Dissection of the conductance of the reduced I_{EI} current of the PING mechanism in a fast (top), slow (middle), and ultra-slow (bottom) timescale.

We can remove these internal I dynamics by replacing the conductance models with LIF models. This should cut away the slow positive feedback. Doing so results in a network that does not (reliably) demonstrate a PING oscillation, as shown in Figure 3.10. Figure 3.10 A shows an example of a raster plot of a PING network with HH neurons in the E population and LIF neurons in the I population, where the maximal conductances in the HH neurons and the leak conductance in the LIF neurons varied by $\pm 20\%$ uniformly. The LIF neurons were set up to match the resting potential and firing threshold of the HH neuron, and otherwise everything was left as in our previous PING simulations. Figure 3.10 B shows how synchrony in a PING network with LIF neurons in the I pool is always less than with HH neurons in the I pool, and decreases faster as parameter uncertainty increases. Here, synchrony is measured via the Spike Time Tiling Coefficient (STTC) [43], and the parameter uncertainty is the fraction by which the conductances (maximal conductances in the HH neuron, leak conductance in the LIF neuron) are allowed to vary uniformly around their nominal value.

We can test our hypothesis that the lack of slow positive feedback is responsible for this by using our population voltage clamp on this new network. The new I_{EI}^{true} is shown in Figure 3.11, A, whilst the new DIC curves are shown in Figure 2.4, B. It can be seen that there is only negative feedback present everywhere now, consistent with the hypothesis that slow positive feedback is required for a robust rhythm.

3.3 Discussion

In this chapter, we investigated the emergence and robustness of PING rhythms in neural circuits. Our primary objective was to elucidate the underlying feedback principles that contribute to the reliability of these rhythms, setting the stage for the exploration of their computational aspects in chapter 4.

In the chapter 2 we argued that animals' neural circuitry will be under evolutionary pressure to perform at the fastest timescales possible. There, we showed that this is not feasible without some mechanism to overcome unreliable spiking, and demonstrated that active dendrites are one way in which neurons can achieve this. We also mentioned how spiking computations in the most rapid regimes effectively put the neurons in a binary regime: any neuron involved can either fire or not fire an action potential during a cycle.

Instead of a mechanism that allows neuron to integrate asynchronous spikes (such as active dendrites), an alternative solution is to increase the synchrony of neuronal firing, which is what we explored via PING rhythms in this chapter. This is a qualitatively different approach: using the PING rhythm, we seek to make the spike timing of neuron more reliable, i.e. reduce jitter, regardless of the specific excitatory current it receives. Indeed, we would

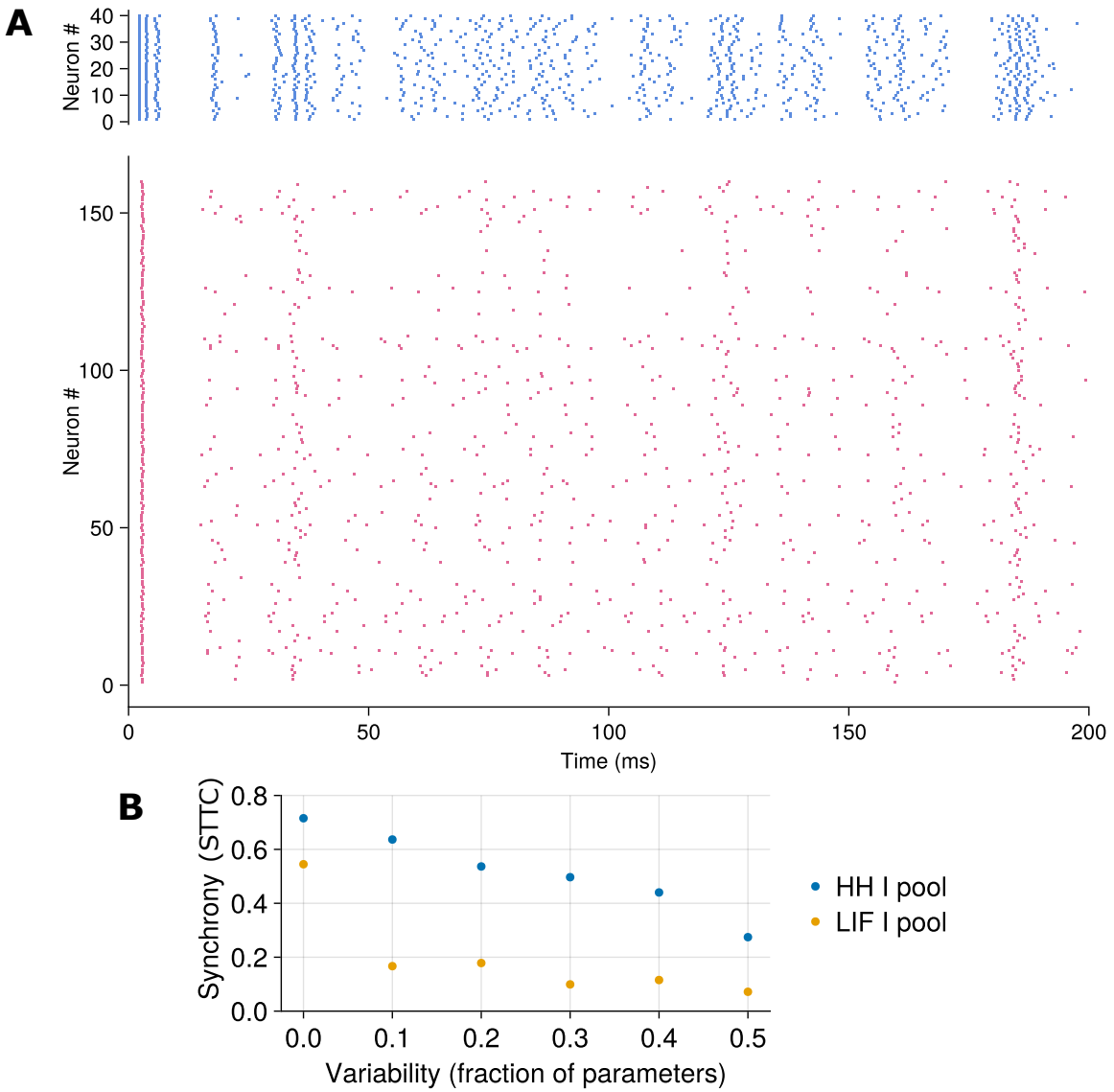


Fig. 3.10 **(A)** An attempt to create a PING rhythm with a recurrently connected population of excitatory HH neurons (spike times shown in pink) and inhibitory LIF neurons (spike times in blue). **(B)** PING synchrony in a network with an HH I pool is higher and more resistant to parameter uncertainty than in a network with an LIF I pool. Synchrony is measured using the Spike Time Tiling Coefficient (STTC) in the E population of HH neurons in two networks; one with HH inhibitory neurons (blue) and one with LIF inhibitory neurons (yellow). Synchrony was measured as the parameters' uncertainty increased uniformly by a fraction around their nominal values.

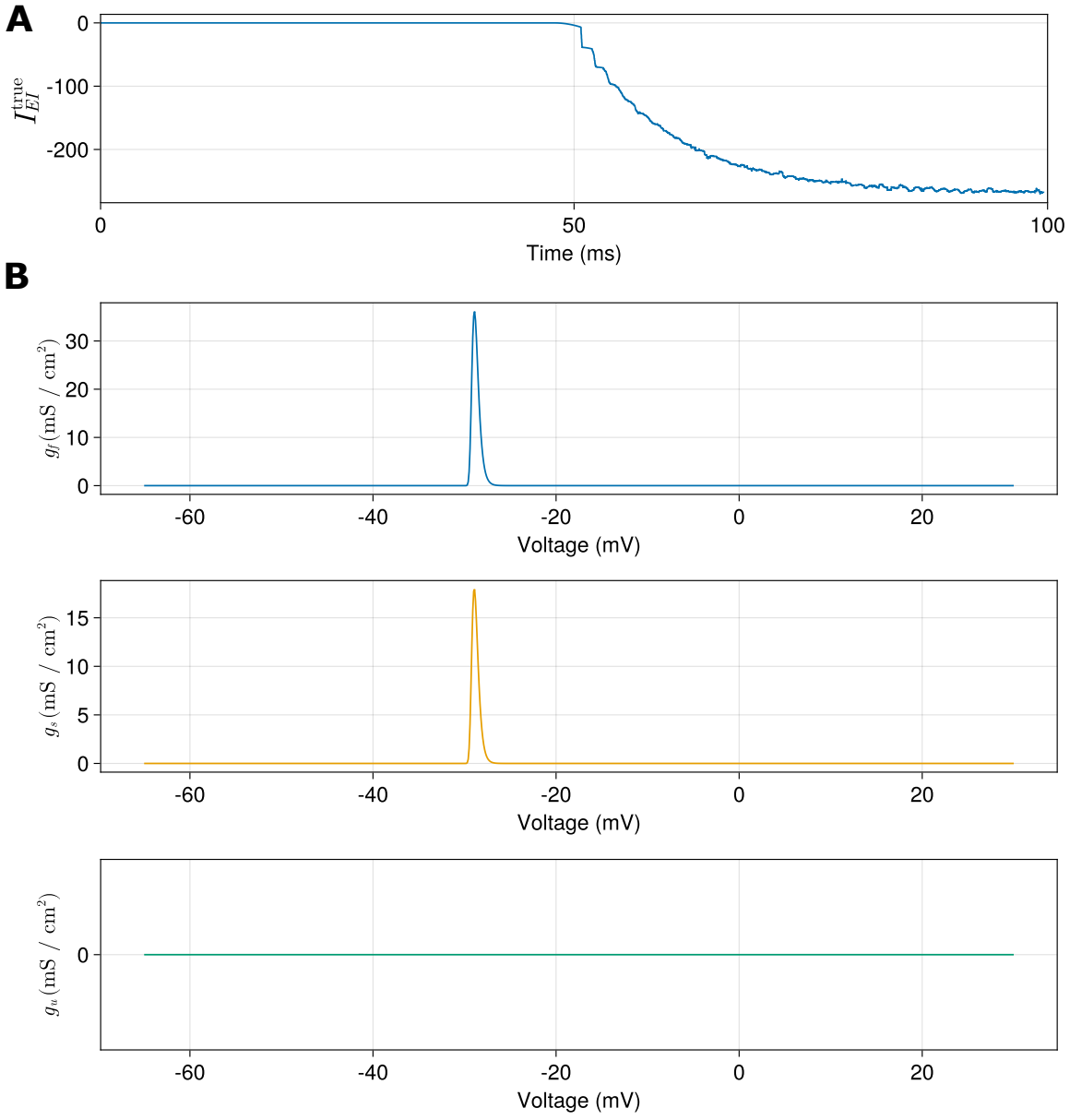


Fig. 3.11 The DIC analysis on a PING network with inhibitory LIF neurons. **(A)** The resulting I_{EI}^{true} after a voltage clamp at $V_1 = 0\,\text{mV}$. **(B)** The computed DIC curves for the I_{EI} with an inhibitory LIF population.

like the spike response to be binary, which we will come back to in chapter 4. However, the two approaches are complementary: active dendrites can work, for instance, as a buffer to store spikes in-between PING cycles, to prevent the unwanted deletion of spikes by the PING inhibition, or it could work as a way of integrating the inputs of multiple PING networks which are operating on different frequencies.

The generic interconnection of numerous neurons in an excitatory-inhibitory (E–I) configuration naturally leads to these PING rhythms [171, 162, 166, 20, 52, 94], and there is experimental evidence that neuronal gamma rhythms, cyclic neural firing patters in the 30 Hz to 100 Hz frequency band, is the result of a PING mechanism [22, 42, 69, 155, 57]. This organization of heterogeneous neuronal behavior into discrete volleys or time-bins potentially enhances the reliability of neural computation. It is important to note that while we focused on rhythm generation at the population level, the specific computational implications of these rhythms will be addressed in chapter 4.

PING changes the behavior of the neurons involved substantially, in addition to synchronizing their activity. The interplay between pools of E and I neurons leads to rapidly fluctuating E and I currents in the neurons participating in the PING, which causes their membrane potentials to fluctuate quickly. This has been associated with reliable and precise firing of action potentials [30, 104, 46, 68]. An excitatory current followed closely in time by an inhibitory current, as is the case in PING, leads to a lower effective spiking threshold, thereby causing the neuron to generate an action potential in response to less net excitatory drive [10], which leads to faster neuron responses. The rapid current fluctuations also decreases the the membrane time constant *in vivo*, because of the increase in conductance due to the synaptic inputs [92, 46, 47]. These features show that PING, in principle, could support fast and precise spiking, which can be exploited for neural computations.

PING rhythms are of particular interest to us due to their temporal correspondence with the spiking computations we aim to implement. The timescale of these rhythms aligns with the fast timescale of the spiking computations that we seek to implement, making them a promising framework for understanding and potentially harnessing neural information processing. In the fastest PING limit, 100 Hz, a neuron will have a 10 ms time window to receive action potentials and fire a spike itself. This is the binary neuron limit that we briefly discussed in chapter 2, where the neuronal outputs can be characterized by 1s and 0s, as each neuron can either produce an action potential (i.e. produce a 1) or not produce an action potential (i.e. produce a 0).

We began by recapitulating the conditions necessary for robust PING coherence in neural circuits. These include strong excitation into the excitatory population, sufficiently strong and dense connectivity from excitatory to inhibitory populations, equally robust connectivity

from inhibitory to excitatory populations, and weak external excitation to the inhibitory population.

A contribution of this work is the introduction of the population voltage clamp, a novel analytical tool that enabled us to derive a mean-field model of PING oscillations. This model is distinctive in its incorporation of both intra-neuronal conductances and synaptic conductances between excitatory and inhibitory populations. This approach offers a more biophysically grounded alternative to traditional Wilson-Cowan type mean-field models [171, 170], providing deeper insights into the biological parameters underlying PING rhythms.

Utilizing this mean-field model and DIC analysis [**drionDynamicInputConductances2015**], we uncovered a surprising feature of the PING rhythm: the presence of slow positive feedback. This finding challenges the naive view of the E–I loop as purely negative feedback due to the involvement of inhibitory neurons. The discovery of slow positive feedback within PING is noteworthy as slow positive feedback has been connected to robust spiking [**franciRobustTunableBursting2017**], neuromodulation allowing neurons to tune in and out of rhythms [51], and general regulation of rhythms through neuromodulation or homeostasis [48]. The identification of this underlying feedback motif provides a tool for predicting whether a given network configuration will exhibit reliable PING oscillations. This understanding has implications for investigating the plasticity and adaptivity of the rhythm, its tunability (relevant for spiking computations), and may allow us to make informed predictions about the effects of neurotransmitters or pharmacological interventions on spiking computation. The question to which extent the mean-field model will be successful in this, is left for future work.

In conclusion, our work provides a novel perspective on the emergence and stability of PING rhythms, emphasizing the role of network dynamics and revealing unexpected feedback mechanisms. These insights not only advance our theoretical understanding of neural oscillations but also offer practical tools for future investigations into neural computation and the development of neuromorphic systems. How PING rhythms specifically come into play for our spiking computation framework, we discuss in chapter 4.

Table 3.1 Default values for simulations with AMPA currents

τ_R	τ_D	g_{AMPA}	E_{GABA_A}
0.2	2	1	-80

Table 3.2 Default values for simulations with GABA_A currents

τ_R	τ_D	g_{GABA_A}	E_{AMPA}
0.5	10	0.5	0

3.4 Methods

3.4.1 AMPA / GABA_A currents

Unless otherwise specified, the values used to simulate the AMPA and GABA_A synaptic currents are given in Table 3.1 and Table 3.2, respectively.

3.4.2 Necessary conditions for PING

For the simulations in Figure 3.3, we used the same architecture as in Figure 3.2, with the following additions:

1. For Figure 3.3A, each neuron in the E population was given an applied current that increased with time:

$$I^E_{\text{app}}(t) = X\frac{t}{200}, \quad X \sim \mathscr{U}(10,14).$$

2. For Figure 3.3B, the synaptic weight of the E $\rightarrow$ I synapses was made a function of time:

$$g_{\text{AMPA}}(t) = \frac{0.7t}{200}.$$

3. For Figure 3.3C, the synaptic weight of the I $\rightarrow$ E synapses was made a function of time:

$$g_{\text{GABA}_\text{A}}(t) = 0.5H(t-100).$$

4. For Figure 3.3D, the cells in the I population were given an applied current in addition to the excitation they receive from the E pool:

$$I^I_{\text{app}}(t) = 2X\,H(t-90), \quad X \sim \mathscr{U}(10,14).$$

3.4.3 Voltage clamp

We worked with Hodgkin-Huxley style neurons (Equation 2.3), in a network with the PING architecture, with a clamping current added to the $\dot{V}$ equation of the excitatory neurons:

$$I_{\text{clamp}} = k(V_i - V), \quad i \in \{0,1\}. \tag{3.21}$$

The voltage clamp would go from $V_0 = -65\,\text{mV}$ to a V_1 which was varied during the simulation. The gain k was set to $k = 800$. The step from V_0 to V_0 was done at $t_{textclamp} = 25\,\text{ms}$. The values used for V_1 are: -60, -35, -30, -28, -26, -24, -20, -15, -10, 0, 10. For every V_1 we would record the average GABA_A current $\langle I_{\text{GABA}_\text{A}} \rangle$ going from the I to the E population. We postulated that this current could be approximated by an equation of the form of an ionic current:

$$\begin{aligned} I_{EI} &= g_{\text{GABA}_\text{A}} \underbrace{m^a h^b}_{:= g_{\text{pop}}} (V - E_{\text{GABA}_\text{A}}) \\ \dot{m} &= (m_\infty(V) - m)\,/\,\tau_m(V) \\ \dot{h} &= (h_\infty(V) - h)\,/\,\tau_h(V), \end{aligned} \tag{3.22}$$

where we chose $a = N_I = 25$ and $b = 1$ to best match the measured timeseries for the conductance.

For every V_1 we found g_{pop} by dividing $\langle I_{\text{GABA}_\text{A}} \rangle$ by $g_{\text{GABA}_\text{A}}\,(V_1 - E_{\text{GABA}_\text{A}})$. For lower values of V_1 the inactivation of the synaptic current was complete (the steady state value was 0). For these values, we set $m_\infty(V_1) = \max(g_{\text{pop}}(t))$. This gave us enough measurement points to fit a curve

$$\begin{aligned} m_\infty(V) &= \sigma(V, \theta) \\ \sigma(V, \theta) &= A\left(1 + \exp\left(-\theta[V - V_{1/2}]\right)\right)^{-1} + \text{ base}. \end{aligned} \tag{3.23}$$

Here, base is the value of $m_\infty(V_0)^{1/N_I}$, $A = \max(g_{\text{pop}}) - \text{base}$, and $V_{1/2}$ is the half-activation value of V_1 for which we saw an inflection point in the measured sigmoid. We fit θ to our datapoints using the Julia package LsqFit.jl [100].

For higher values of V_1, g_{pop} displayed oscillatory behaviour (Figure 3.4, right). When this occurred, we fit an analytical curve from the first peak of g_{pop} to its second peak, following the form

$$g_{\text{pop, fit}}(t, \theta) = \alpha + \beta \exp\left(-\theta[t - t_{\text{clamp}}]\right), \tag{3.24}$$

where $\alpha = \min_{t > t_{\text{clamp}}}\left(g_{\text{pop}}(t)\right)$, $\beta = g_{\text{pop}}(t_{\text{clamp}})$, and θ was again fitted using LsqFit.jl.

Our inactivation variable h is qualitatively different from its Hodgkin-Huxley counterpart: similarly to sodium inactivation, it inactivates slowly at higher voltages. For small depolariz-

ing steps, where m_∞ is non-zero, the steady-state conductance is always zero (Figure 3.4, left). Therefore we chose to set $h_\infty(V_1) = 1 - m_\infty(V_1)$ for these low V_1 values. But, because voltage clamps at higher voltage values lead to a non-zero steady state conductance (Figure 3.4), the value of h_∞ should be non-zero at high voltages. This means that h_∞ is not a monotonically decreasing function, which is questionable from a biophysical viewpoint. To find h_∞ for the high voltage values where it is non-zero, we first found the range of V where the steady-state value of g_{pop} was non-zero. For these curves, we used the analytical fit of Equation 3.24 to find $h_\infty(V_1)$. We could assume safely that for these values $m_\infty(V_1) = 1$, as the sigmoid of m_∞ had well reached its saturation regime here. Then we found $\lim_{t\to\infty} g_{\text{pop, fit}}(t) = h_\infty(V_1)$. Note that $h_\infty(V)$ saturates at 0.2, not at 1 as per usual for (in)activation variables. Ultimately we find

$$h_\infty(V) = (1 - m_\infty(V)) + a\left(1 + \exp(-b[V - V_{1/2}])\right)^{-1}, \tag{3.25}$$

with $A = 0.2$ being the maximum non-zero conductance for large depolarization steps, $V_{1/2}$ the observed half activation value at those values where $m_\infty \approx 1$, a was the maximum non-zero steady-state value of g_{pop} at high voltages, and b was fit to the measured datapoints using LsqFit.jl.

It is worth noting that the non-zero values of $h_\infty(V)$ for $V \gg 0$ can be safely ignored, and that h_∞ can be approximated as a monotonically decreasing function, because $\tau_h \gg \tau_m$.

Chapter 4

A framework for reliable SNNs

4.1 BNN interpretation of PING rhythms

We have spent some time now talking about PING rhythms, but why should we care about it? There are a few noteworthy features present in the PING mechanism:

1. The short period of the PING oscillation permits each E neuron to integrate at most one spike from each of its inputs before its memory gets 'reset' by the wave of inhibition from the I pool;

2. Analogously, each neuron has only enough time to make one firing decision: it either fires or it does not;

3. In this way, activity propagates through layers so that each layer produces a volley where each neuron outputs at most one spike.

These three features imply that a PING network can operate in a binary regime, a notion we briefly discussed already in chapter 2: each neuron has just enough time to receive one spike from each of its inputs, and then it can fire at most one action potential. When we chain several layers of excitatory neurons together, as shown in Figure 4.1A, the PING mechanism will ensure that each layer fires synchronously as activity is propagated through the network (Figure 4.1B). Then we can always make the analogy with a binary neural network, where each neuron in the SNN corresponds to a unit in the BNN. In a BNN, each unit's output is either a 1 or a 0. Similarly, in the PING network, during one PING cycle, each neuron either fires or it does not. Indeed, we can always train a feedforward BNN to mimic the output of the feedforward PING SNN with the architecture of Figure 4.1A, provided that we interpret the spiking output of the PING network in this binary manner. The converse does generally not hold: not every BNN network activity can be reproduced by a PING circuit, for example

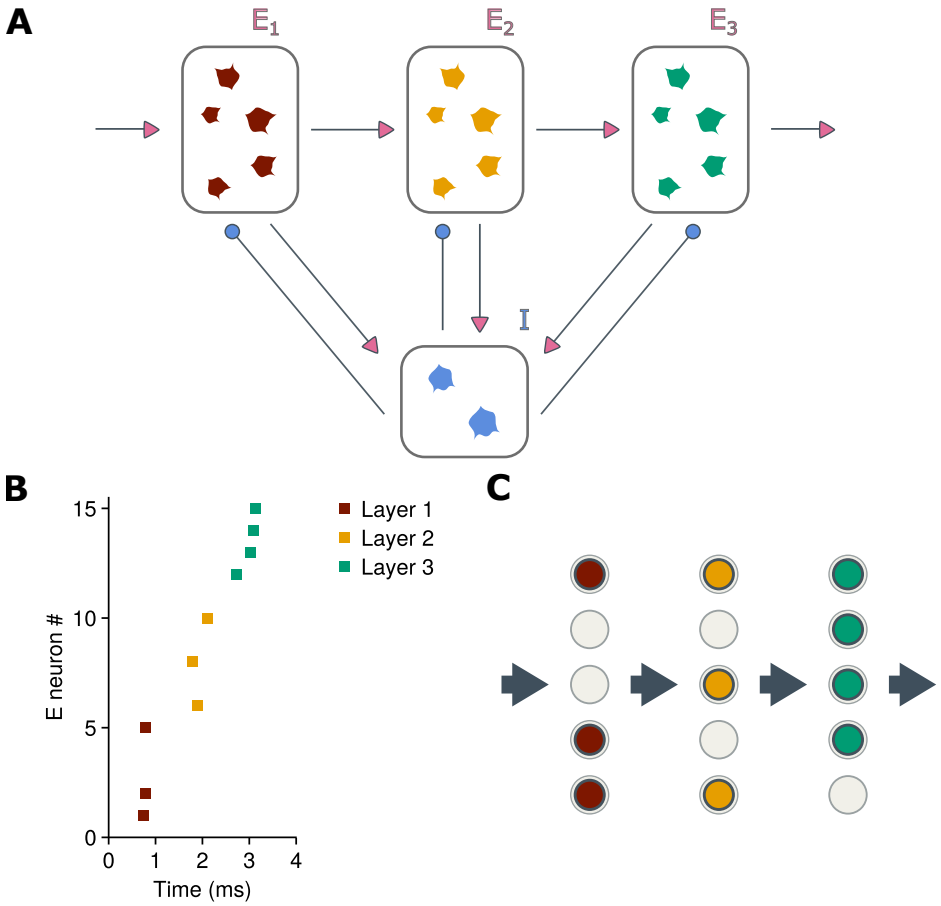


Fig. 4.1 Illustration of how a PING network can mimic a BNN. **(A)** Sketch of a feedforward network consisting of several layers of excitatory (E) layers, all recurrently connected to a single pool of inhibitory (I) neurons. **(B)** The raster plot of the network from **A**, showing the spiketimes of the neurons in the three layers for one PING cycle. **(C)** The binary interpretation of the activity in **B**: in each PING cycle, each neuron in each layer is either 'on' (it fires a spike which is transmitted to the next layer), shown here as a filled circle, or it is 'off' (it does not fire a spike), shown here as an empty circle.

because the total number of 1s (spikes) is too low to excite the I pool and thereby maintain a PING rhythm. This could be achieved by, for example, adding an extra E pool that always excited the I pool, even in the absence of inputs, but whether that is desirable or not we will not address here.

In effect, each neuron in one layer of the PING network then acts as a coincidence detector of spike volleys coming from the previous layer. Whether the neuron reaches its threshold or not is determined by the synaptic weights and by the degree of synchrony of the incoming spike volley. Each layer therefore receives a volley of synchronous input spikes, by virtue of the E–I interconnection, and therefore also produces a volley of synchronous output spikes.

4.2 PING circuits can compute

With this architecture in hand, there are a few tests we can do to see if this network can compute in principle.

For a computation to be able to occur, the rhythm must be robust and stable, whilst the identity of the neurons participating in each cycle must be flexible. After all, just as with ANNs, the identity of the neurons active during a PING cycle is what sends information through the circuit.

To test this, we set up a two layer network, with 50 neurons in each layer, but only the second layer having an $E--I$ connection to a pool of inhibitory neurons. The first layer was made to produce asynchronous spikes. These spikes would affect layer two via AMPA synapses, with a weight matrix W, where each weight element w was drawn from a distribution $w \sim \mathcal{N}(\mu, \sigma)$. That is, the synapses follow the dynamics of Equation 3.1, with $g_{\text{AMPA}} = w$.

We computed the participation and coherence of the neurons in layer two. Participation meant the fraction of neurons that would fire at least one spike. Coherence was defined as the average pairwise cross-correlation of the firing times of the neurons. Let $S_i(t) = \sum_{t_s} \delta(t - t_s)$ be the spiketrain of neuron i, where t_s is defined as the timepoint where its voltage had an upwards crossing of 0. Then the pairwise cross-correlation is

$$C_{ij}(\tau) = \int_{-\infty}^{\infty} (S_i * G)(t) \cdot \big(S_j * G\big)(t+\tau)\, dt, \tag{4.1}$$

where $G(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-t^2/2\sigma^2\right)$.

The participation and coherence of this readout layer are shown in Figure 4.2. The coherence of the rhythm is strong for almost all levels of participation. This is to be expected,

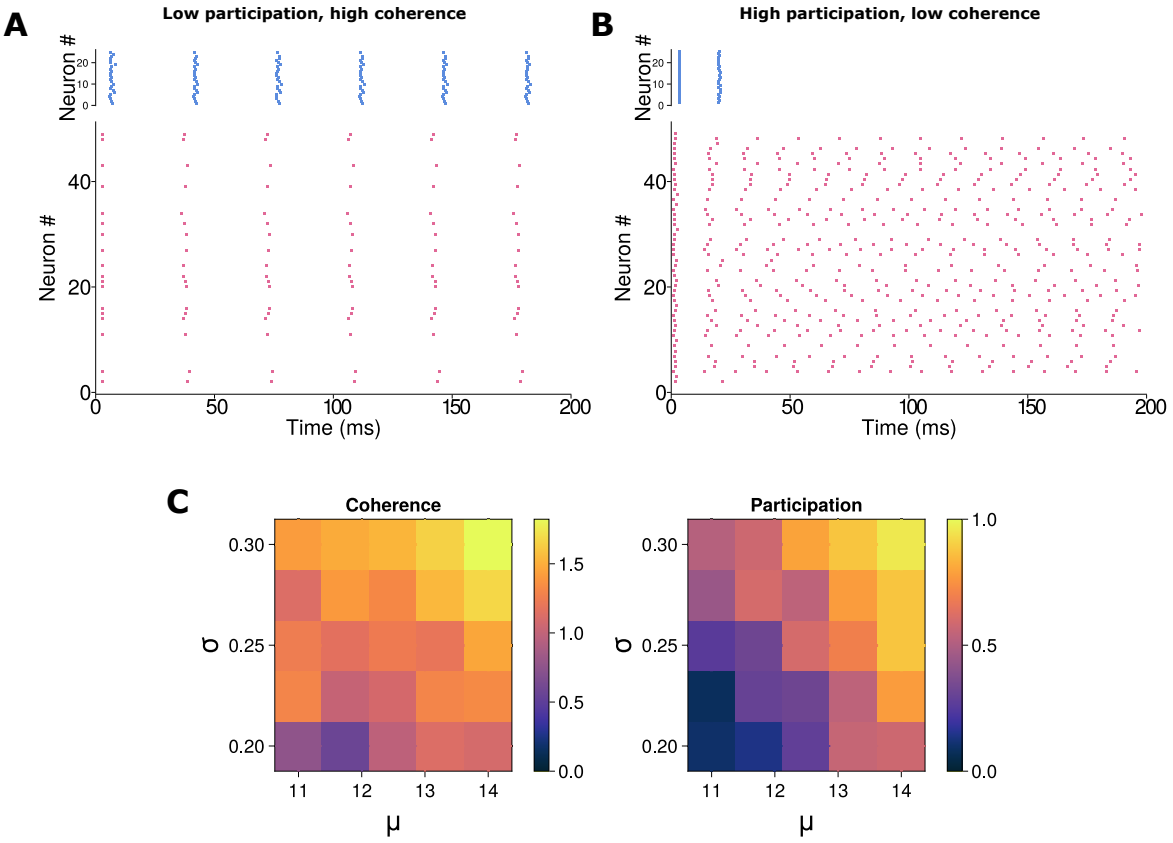


Fig. 4.2 Variable levels of neuron participation in PING support coherence rhythms. **(A)** Example of a PING with low participation (~1/3 of neurons participating), but high coherence. **(B)** All neurons participate, but there is no coherent rhythm. **(C)** Coherence and participation levels of the readout layer of a two-layer PING circuit, for varying means μ and variance σ of the connecting weight layer.

as we showed in the previous chapter that PING rhythms have a threshold of E population activity where it kicks in. This implies that, provided that at least this threshold fraction of E neurons fires every cycle, the rhythm will be present. Therefore, neurons can 'drop in and out' of the computation, meaning that the network is flexible enough to handle different inputs and outputs.

To test if PING like circuits as described above could perform a computation in principle, we measured the mutual information between input and output spikes in a PING network and an asynchronous network. Both networks have identical architectures; one input layer, two hidden layers, and an output layer, with each layer consisting of 10 neurons. Both have one pool of inhibitory neurons; for the asynchronous network these are LIF neurons, as in Figure 3.11.

The inputs are binary vectors, and each network was given a time window to settle on an answer. The answer was interpreted as each neuron in the output layer that fired at least one spike during the time window. Both networks were randomly parameterised; that is, equipped with weight matrices for the weights between layers which were randomly generated.

For $N_{IO} = 500$ randomly generated input vectors, this procedure gave N_{IO} output vectors as well. We computed the mutual information between the sets of input and output vectors (Figure 4.3C). The mutual information in the inputs and outputs of the PING network quickly saturated as a function of the time window. For the asynchronous network, as expected, the spikes of different samples interfered with each other, and the outputs were not consistent for the inputs. This suggests that for quick computations, synchronous networks are more robust.

It must be said that mutual information in input and output and computational capacities of an input-output system are not the same. For instance, mutual information does not capture the complexity of the computations performed; an input-output system that implements the identity map (leaves the input unchanged) has maximal mutual information, but has very low computational complexity. It does however give us an upper bound on the amount of useful computations that a system can perform: a system cannot compute more information than it processes. From this we can conclude that a PING network, at least, has the capacity for more computations, especially on short timescales, than an asynchronous network.

4.3 Training SNNs using the teacher forcing algorithm

4.3.1 Feedback sensitivity

Training SNNs is a very hard problem, primarily due to the gradients that are not well-behaved. Loosely speaking, a neuron is only sensitive to parameter changed when close to its spiking threshold, and insensitive everywhere else in its state-space. Imagine using a gradient-based method to train a network of spiking neurons. If a neuron is not active, changes in parameter will have no or very little effect on the neuron's outputs, and therefore the gradients will vanish. Conversely, if a neuron is close to threshold it will be very sensitive to parameter changes, and gradients will explode.

We can make this more precise by using a linearised neuron model, following a methodology which is very similar to what we did in section 3.2.2, but incorporating the forward dynamics (i.e. the membrane integration) also this time, following [55]. We use the, by now

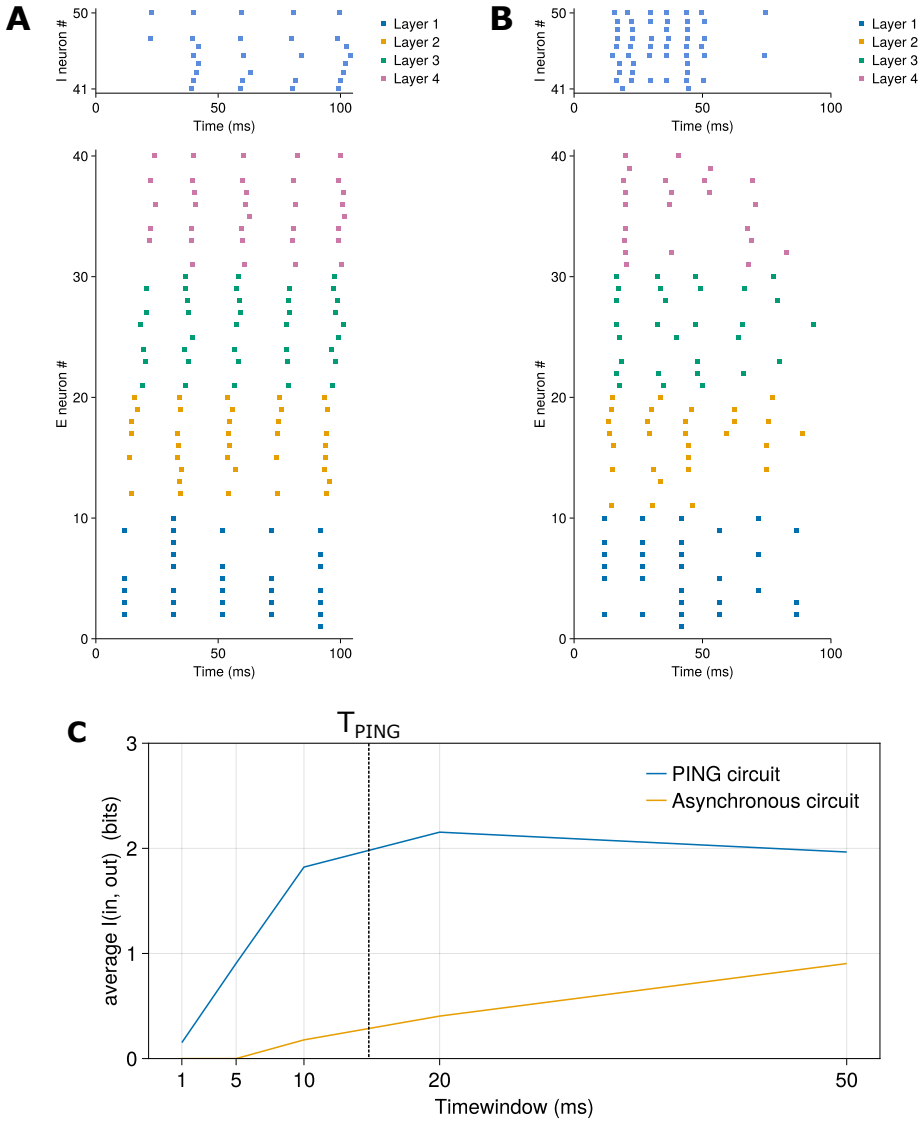


Fig. 4.3 PING networks preserve information of the inputs. **(A)** Example of input-output mappings of a PING network. **(B)** Example of input-output mappings of a network with the I pool replaced by LIF neurons. Note that the synchrony starts to disappear eventually. **(C)** Mutual information between inputs and output of the PING and the asynchronous networks, as a function of the timewindow between inputs. The information transmission of the PING saturates around the PING time period.

very familiar, equation for the dynamics of the membrane potential

$$\begin{aligned} C\dot{V} &= -I_m(t,V) + I_{\text{app}} \\ &= -\sum_i g_i(t,V)(V-E_i) + I_{\text{app}}. \end{aligned} \tag{4.2}$$

We interpret this in the language of control theory as a forward transfer function

$$P(s) = \frac{1}{Cs} \tag{4.3}$$

which represents the passive membrane integration dynamics, and a negative nonlinear feedback connection with input V and output I_m.

As we already know, the channel dynamics are usually modelled by introducing gating variables. Let these gating variables be a vector $x \in \mathbb{R}^N$ such that

$$\begin{aligned} g_i(t,V) &= g_i(x(t,V)) \\ \tau_i(V)\dot{x}_i &= -x_i + x_{i,\infty}(V). \end{aligned} \tag{4.4}$$

We can now linearise around the quasi-steady state $(V, x_\infty(V)) \coloneqq (V, x_{1,\infty}(V), \ldots, x_{N,\infty})$:

$$\begin{aligned} C\delta\dot{V} &= -\frac{\partial I_m}{\partial V}\delta V - \sum_{i=1}^{N} \frac{\partial I_m}{\partial x_i}\delta x_i + \delta I_{\text{app}}(t) \\ \tau_i(V)\delta\dot{x}_i &= \frac{\partial x_{i,\infty}}{\partial V}\delta V,\ i = 1,\ldots,N, \end{aligned} \tag{4.5}$$

and take the Laplace transform:

$$\begin{aligned} Cs\widetilde{\delta V} &= -\frac{\partial I_m}{\partial V}\widetilde{\delta V} - \sum_{i=1} N\frac{\partial I_m}{\partial x_i}\widetilde{\delta x_i} + \widetilde{\delta I}_{\text{app}}(s) \\ (\tau_i(V)s+1)\,\widetilde{\delta x_i} &= \frac{\partial x_{i,\infty}}{\partial V}\widetilde{\delta V},\ i = 1,\ldots,N, \end{aligned} \tag{4.6}$$

where $\tilde{\cdot}$ indicates the Laplace transform. We have a feedforward transfer function $P(s) = \frac{1}{Cs}$ and a feedback transfer function

$$H(s,V) = g_0(V) + \sum_{i=1}^{N} \frac{g_i(V)}{\tau_i(V)s+1}, \tag{4.7}$$

where

$$g_0(V) = \left.\frac{\partial I_m}{\partial V}\right|_{(V,x_\infty(V))}, \quad g_i(V) = \left.\frac{\partial I_m}{\partial x_i}\right|_{(V,x_\infty(V))} \left.\frac{\partial x_{i,\infty}}{\partial V}\right|_V. \tag{4.8}$$

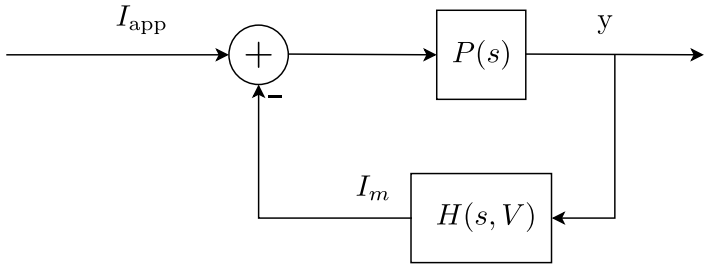


Fig. 4.4 A block diagram of the feedback interconnection of the dynamics inside a generic (linearised) neuron model, which takes as input an applied current I_{app} and produces output voltage y via the feedforward transfer function $P(s)$. The dynamics are shaped by the feedback effect of the ion channels in the membrane of the neuron, who take as input the output voltage y and produce a set of membrane currents I_m, via the feedback transfer function $H(s,V)$. The V argument in $H(s,V)$ emphasises that this transfer function depends on the voltage value around which the model is linearised.

Together $P(s)$ and $H(s,V)$ form the feedback interconnection depicted in Figure 4.4. From classical control theory [7, chapter 8], we know that the loop gain of this feedback loop is $L(s,V) = P(s)H(s,V)$, that is

$$L(s,V) = \frac{1}{Cs}\left[g_0(V) + \sum_{i=1}^{N} \frac{g_i(V)}{\tau_i(V)s+1}\right]. \tag{4.9}$$

The condition for ultra-sensitivity, where the neuron model reaches threshold [55], is then $L(i\omega,V) = -1$, for $\omega \in \mathbb{R}^+$ [7]. It is at this point that the transfer function of the system with feedback has a pole that lies on the imaginary axis, and the eigenvalue of the interconnected system becomes non-negative.

In addition to finding where the threhold of the neuron model is, by leveraging the ultra-sensitivity condition, we can also compute how sensitive (or insensitive) the neuron model is elsewhere. We can do this by computing how close $L(i\omega,V)$ gets to -1 for a given voltage V. That is, we compute

$$\max_{\omega\in\mathbb{R}^+} \left|\frac{1}{1+L(i\omega,V)}\right|, \tag{4.10}$$

where we have taken the inverse so that larger values of the quantity correspond the sensitivity, and lower values to insensitivity, which is the intuitive metric. Lower values imply that variations in parameters in the model have a small effect on the output voltage of the neuron, and vice versa for larger values. In particular, the divide between attenuation and amplification of such variations lies at 1.

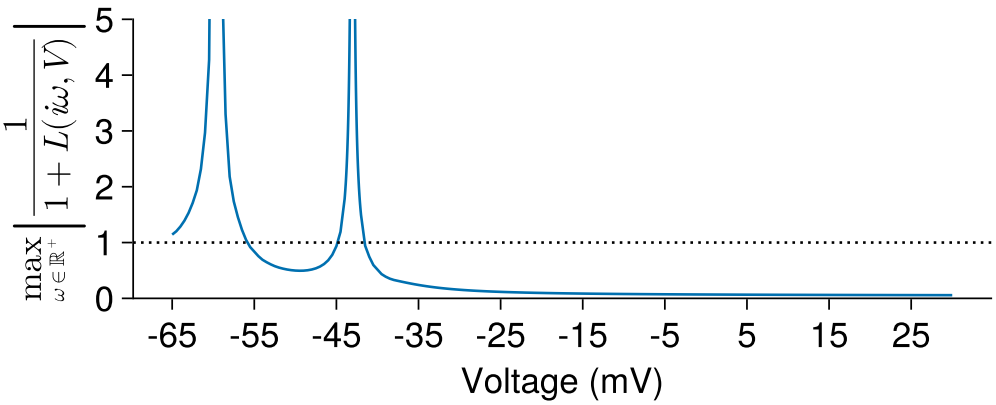


Fig. 4.5 The amplification of inputs in the linearised Hodgkin-Huxley model. The dotted line is where the magnitude of the loop gain function equals 1; above this line there is amplification, below this line there is attenuation of inputs.

We have plotted $\max_{\omega \in \mathbb{R}^+} |1/(1+L(i\omega,V))|$ for the Hodgkin-Huxley model in Figure 4.5. It is immediately apparent that there are two narrow voltage regions of high sensitivity, and that elsewhere the sensitivity is very low. This matches the intuition that in any system equipped with thresholds, the output will only be sensitive to the input when close to threshold, and when away from threshold the system's output remains unaffected by variations in input.

This fundamental property of excitable systems must be taken into account when training SNNs. One naïve way of training SNNs would be to try and use stochastic gradient descent to optimise some cost function directly on the network. Then updates to weight w_{ij}, the weight from neuron j to neuron i, during learning will contain terms such as $\frac{\partial V_i}{\partial w_{ij}}$. The weight w_{ij} is connected to neuron i via a synaptic current I_{syn}. Let us assume that we can treat I_{syn} as the I_{app} in the above. Then we have $\frac{\partial V_i}{\partial w_{ij}} = \frac{\partial V_i}{\partial I_{\text{syn}}} \frac{\partial I_{\text{syn}}}{\partial w_{ij}}$, and the preceding argument suggests that $\frac{\partial V_i}{\partial I_{\text{syn}}}$ (which is the sensitivity of the output with respect to an input) is not well-behaved for almost all values of V, leading to either exploding or vanishing gradients which are very detrimental to learning.

There are ways around this, such as instead of taking a derivative of an output taking the derivative of an output that is averaged over a sufficiently long period of time, or otherwise smoothing or approximating neural outputs with techniques such as using surrogate gradients [80, 172, 173]. These methods work well for networks consisting of LIF neurons, which depend only on their past through an instantaneous reset when the LIF neuron spikes. However, it remains to be shown that such methods can also be applied succesfully to networks of neurons which depend on the history of their activity in more complicated

ways, in multiple timescales. In the next section we show that making these methods work for conductance-based neurons, which have several dynamic components which evolve in different timescales, is not straightforward.

4.3.2 Surrogate gradients and conductances

Surrogate gradient methods have seen a surge of interest in the last few years as a means of training SNNs [117, 173, 172]. These methods rely on smoothing out the instantaneous reset of LIF neurons, which models their spiking.

Therefore, we first rewrite the LIF dynamics (Equation 2.11) in a form that explicitly incorporates the Heaviside function as the spiking output. The membrane potential dynamics are still

$$\tau \dot{V}_i = -V_i + I_{i,\text{syn}} \tag{4.11}$$

and we write for the output spike train produced by neuron i

$$S_i(t) = \vartheta H\left(V_i(t) - \vartheta\right), \tag{4.12}$$

where, as before, ϑ is the threshold potential. Then

$$\dot{I}_{i,\text{syn}} = -\frac{I_{\text{syn}}}{\tau} + \sum_{j \in \mathscr{I}} w_{ij} S_j(t). \tag{4.13}$$

where $\mathscr{I}$ is the collection of indices of the upstream neurons to which neuron i is connected.

To make the connection with ANNs, we discretise the dynamics in time. We use the exponential Euler method for this (which we also used in section 2.1.2) [40]. For a simulation stepsize of Δt this yields

$$\begin{aligned} V_i[t+1] &= \alpha V_i[t] + I_i[t] - S_i[t] \\ S_i[t+1] &= \vartheta H\left(V_i[t] - \vartheta\right) \\ I_i[t+1] &= \alpha I_i[t] + \sum_{j \in \mathscr{I}} w_{ij} S_j[t], \end{aligned} \tag{4.14}$$

with constant $\alpha = e^{-\Delta t/\tau}$.

Note that the reset of this neuron model is given by the $-S_i[t]$ term in $V_i[t+1]$; when $V_i[t] = \vartheta$ then $S_i[t] = \vartheta$, which results in $V_i[t+1] = V_i[t] - S_i[t] + I_i[t] = \vartheta - \vartheta + I_i[t] = I_i[t]$. Furthermore, note that this system can be seen as a deep feedforward ANN with units of $V_i[t], I_i[t], S_i[t]$, where every timestep t forms a layer of the network.

The point is that when we try to optimise a loss function $\mathscr{L}$ which depends on the output of this network, we calculate weight updates $\Delta w_{ij}[t]$ which reduce this loss function. These

terms will depend on the output of each neuron

$$\Delta w_{ij} \propto H'\left(V[t] - \vartheta\right) \frac{\partial V_i[t]}{\partial w_{ij}}, \tag{4.15}$$

but this will impede the flow of error signals in the backwards pass: because $H'(x)$ is 0 for all $x \neq 0$, and ill-defined at 0, the Δw_{ij} terms would always evaluate to zero, and training would not occur. The idea behind surrogate gradient methods is therefore to replace the $H'(V_i[t] - \vartheta)$ in Δw_{ij} and simply replace it by something smooth, specifically by the gradient of a function which still approximates the step function.

To demonstrate the effectiveness of surrogate gradients, we use a simple synthetic dataset, taken from [149]. The dataset consists of 100 random spiketrains, where every spiketrain consists on average of 5 Hz spikes. For details and an example of one collection of input spiketrains, see section 4.6.2.

In addition to the 100 input units, the network is equipped with 4 hidden and 2 output units. The neurons in the hidden layer have a membrane time constant of $\tau = 10\,\text{ms}$, and the neurons in the output layer have a time constant of $\tau = 20\,\text{ms}$.

The goal of the network was to perform a binary classification on this set of spike patterns, with the labels being randomly created. To this end, the output units were not equipped with a firing threshold, so they just performed a leaky integration. Their outputs used in the cost function were

$$\hat{V}_i^{\text{out}} = \max_t \left(V_i^{\text{out}}[t]\right),$$

and these values were used in the crossentropy function, as we did before in section 2.1.2.

We attempted to train this network twice (Figure 4.6). First we used the real gradient during training, using $H(x)$ as the spiking function. This prevented the network from being able to do the classification, resulting in a quick plateauing of the loss during training, and ultimately in a a 53% classification accuracy, or just above chance. The reason for this just above chance performance is that the readout neurons can still be trained (as they are not equipped with spiking thresholds), so they can contribute a little bit to the performance. It turns out that the outputs of the four hidden layer neurons did not change during training; because the weight updates are zero due to vanishing gradients.

To get the hidden layer to participate, we use a smooth approximation of the step function, which has a well-defined threshold (Figure 4.6A). We used the SuperSpike derivative, introduced in [172], which is the derivative of the fast sigmoid function. This simple change allows the weights to the hidden layer to be updated, and this results in the good classification performance of 96%.

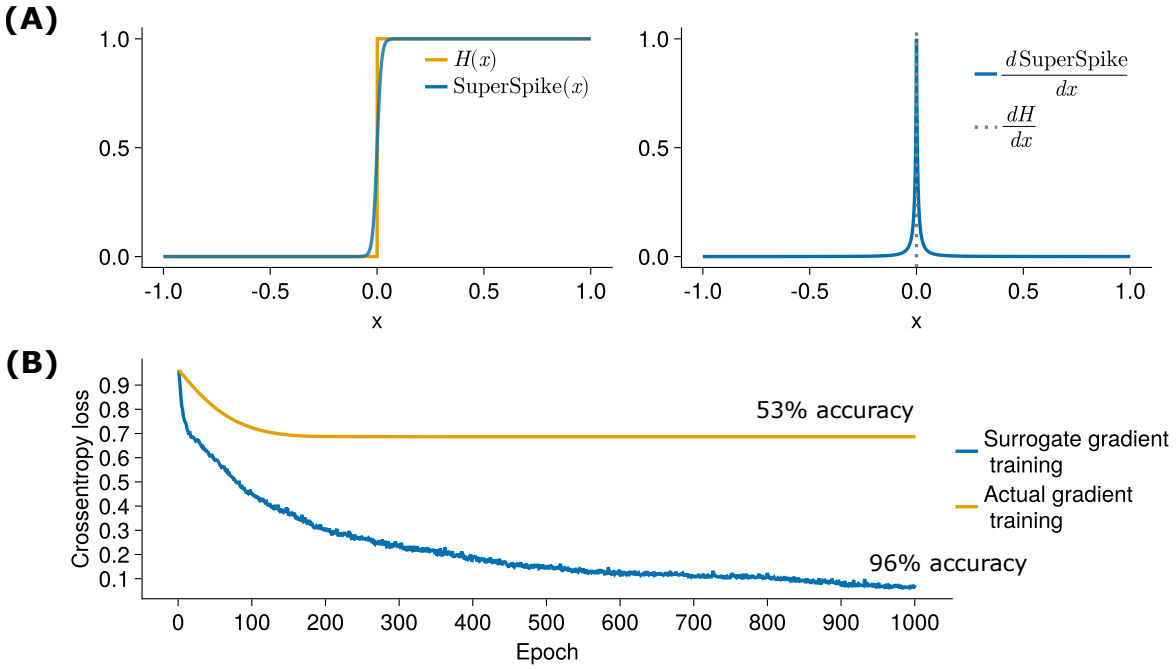


Fig. 4.6 Training an LIF-based SNN using surrogate gradients. **(A)** (left) The actual spiking function $H(x)$ and its approximation SuperSpike(x) whose gradient is used for training. (right) The actual gradient of the spiking function is ill-defined, but the SuperSpike gradient is nonzero and therefore allows for optimisation to occur. **(B)** The decrease of the total loss during training when using the real gradient and the surrogate gradient.

To see how well surrogate gradient methods transfer to more biophysically plausible neuron models, where the subthreshold dynamics are not necessarily linear, we attempted to implement them in a neuron model that still had a hard spiking threshold that could be modelled with the Heaviside function H, but also had nontrivial subthreshold dynamics.

A good candidate for such a model is the Multi-Quadratic Integrate-and-Fire (MQIF) model [159]:

$$\begin{aligned} C\dot{V} &= g_f\left(V-V^0\right)^2 - g_s\left(V_s-V_s^0\right)^2 + I_{\text{app}} \\ \tau_s\dot{V}_s &= V - V_s \\ &\text{if } V \geq V_{\max}: \\ &\qquad \text{set } V = V_r \\ &\qquad \text{set } V_s = V_{s,r}, \end{aligned} \tag{4.16}$$

where we write the resets in the usual way, without a reference to Heaviside functions as in Equation 4.12. This model is capable of displaying features found in real neurons, such as bistability between tonic spiking and quiescence, spike latency, and bursting [159]. To make the comparison with the LIF model a fair one we rescaled the weights so that the average number of spikes in both networks was the same, and we rescaled our units so that the resting potential of the MQIF model was 0 and its threshold 1 (see section 4.6.3 for the details).

We compute the exponential Euler update rule for this model:

$$\begin{aligned} V[n+1] &= \left(V_\infty\left(V[n],s[n]\right) + \beta_V\left[V[n]-V_\infty\left(V[n],s[n]\right)\right]\right)\left[1-H\left(V[n]-V_{\max}\right)\right] \\ &\quad + H\left(V[n]-V_{\max}\right)V_r \\ V_\infty(V,V_s) &:= \frac{g_f\left(V^2+V^{0^2}\right) - g_s\left(V_s-V_s^0\right)^2 + s[n]}{2g_fV^0} := f(V,V_s) + g(s[n]), \end{aligned} \tag{4.17}$$

where f and g are given by

$$\begin{aligned} f(V,V_s) &= \frac{\bar{g}_f\left(V^2+V^{0^2}\right) - \bar{g}_s\left(V_s-V_s^0\right)^2}{2\bar{g}_fV^0} \\ g(s_n) &= \frac{s_n}{2\bar{g}_fV^0}, \end{aligned} \tag{4.18}$$

see section 4.6.3 for the complete derivation.

We use the same task and same network architecture as we used with the LIF network. We make the resets differentiable exactly as we did before: we replace $H'(x)$ with the SuperSpike derivative for the purposes of training. The decrease of the loss function during training with and without a surrogate gradient is shown in Figure 4.7. It is evident that the results do not

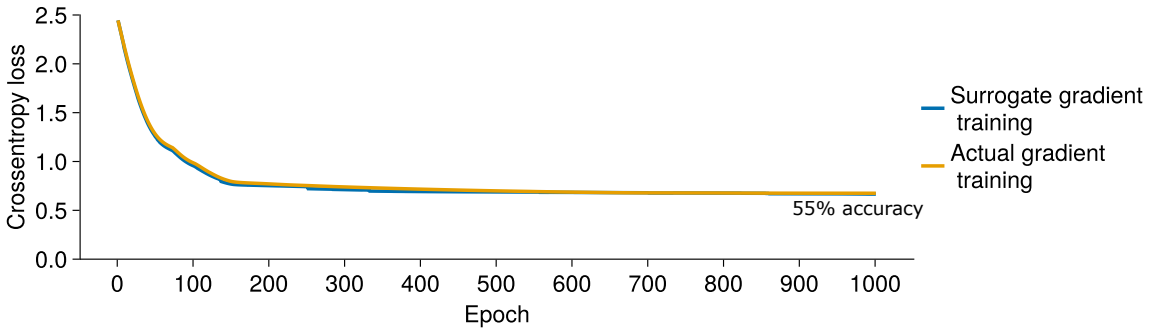


Fig. 4.7 Training progression of the MQIF SNN with and without surrogate gradients. Note that the the blue line (training progression with surrogate gradients) is nearly invisible because the yellow line lies on top of it.

differ; indeed, both networks give the same performance of 55%, a performance just better than randomly guessing, comparable to the performance of the LIF network when trained without surrogate gradients.

To investigate why the surrogate gradient method did not work with the MQIF model, we can compare its voltage update step with that of the LIF network. By rearranging the terms in Equation 4.17 we find that

$$\begin{aligned} V[n+1] = & \underbrace{f(V,V_s)\,[1-\beta_V]+\beta_V V[n]}_{\text{feedback}} \\ & + \underbrace{g(s_n)\,[1-\beta_V]}_{\text{feedforward}}[1-H\,(V[n]-V_{\max})]+H\,(V[n]-V_{\max})\,V_r, \end{aligned} \tag{4.19}$$

where we have marked the terms that represent the feedforward inputs from upstream neurons and the feedback effects on the membrane potential. We can compare this with the LIF update step:

$$V[n+1] = (\underbrace{\beta V[n]}_{\text{feedback}} + \underbrace{s[n]}_{\text{feedforward}})\,(1-H(V[n]-1))\,. \tag{4.20}$$

We see that there is a discrepancy in the feedforward synaptic dynamics, which can be redressed by rescaling the weights in the MQIF SNN (section 4.6.3). But the discrepancy in the feedback effects can not be reconciled so easily, It is this discrepancy that accounts for the difference in learning performance between the two networks. To be more precise: any spiking system which for its spiking output relies on having a region of hypersensitivity shaped by positive feedback (i.e. a threshold), in the midst of a region of very low sensitivity through negative feedback, that is, a dynamical excitable system [141], will encounter

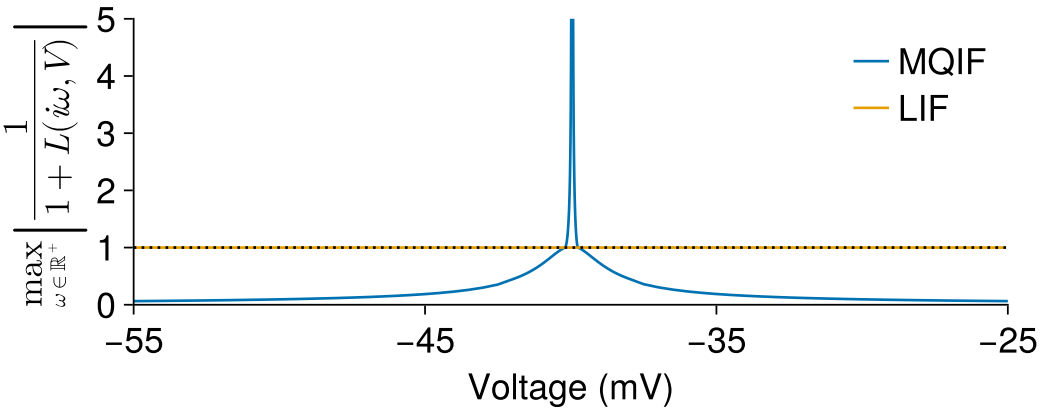


Fig. 4.8 The amplification of inputs in the MQIF and LIF models. The MQIF model possesses a true narrow region of hypersensitivity (i.e., a threshold), surrounded by regions of insensivitity, whereas the sensitivity of the LIF model is uniform and therefore voltage independent. The dotted line is where the magnitude of the loop gain function equals 1; above this line there is amplification, below this line there is attenuation of inputs.

problems with vanishing and exploding gradients during training, thereby harming the performance of the network post-training.

To confirm this, we repeated the analysis of section 4.3.1 for the MQIF and LIF model. For how these sensitivities are computed, see section 4.6.1. The feedback sensitivity of these models is shown in Figure 4.8. As as expected, the sensitivity of the MQIF model shows that the model possesses true excitability, i.e., it has one firing threshold, which is the narrow peak in the centre of Figure 4.8. The LIF model, on the other hand, has uniform sensitivity, and the only way to obtain the spiking behaviour is through the reset. Although such a reset is also present in the MQIF model, but without removing the connection to biophysical conductance-based models; for details see [159].

This corroborates the analysis of section 4.3.1. It is this well-behaved sensitivity of the LIF model that makes it suitable for surrogate gradient methods. Other neuron models, however, such as the MQIF network (even with surrogate gradients), show excitable dynamics, which can mess with the training using surrogate gradients, and thereby lead to poor SNN performance. In the next section we will discuss an alternative training approach which we have developed, that circumvents these difficulties.

4.3.3 Teacher forcing

The teacher forcing method is based on the online model estimation method introduced in [34]. This method relies on the linear parameterization of neuron models in the maximal conductances. To highlight this, we rewrite the conductance model as Equation 2.1 in the

more abstract form

$$\begin{aligned} \dot{V} &= \phi(V,w)^T g + I_{\text{app}} \\ \dot{w} &= f(V,w). \end{aligned} \tag{4.21}$$

Now V is the vector of voltages which we seek to estimate, w is the vector of gating variables on which V depends, and g is the vector of maximal conductances.

Suppose we seek to answer the question "given V and I_{app}, can we find g?". Let $\hat{V}$ be the dynamics of our estimated 'observer' model, that is, the model that depends on $\hat{g}$, our guess of g. We will use the loss function

$$L(g) = \int_0^T e^{-\alpha\tau} \left(\bar{H}\dot{V} - \bar{H}\dot{\hat{V}}\right)^2 d\tau, \tag{4.22}$$

where T is the time period over which you take measurements and α is a constant forgetting factor. Here $\bar{H}$ is some linear filter, that filters out the noise in the measurements. This is very important to ensure the whole method works in practice; without the filtering the method is very sensitive to noise, and does not converge. A good choice for $\bar{H}$ is $\bar{H} = \frac{\gamma}{s+\gamma}$. $\bar{H}$ needs to be positive real (or 'passive'), i.e. $\text{Re}\{\bar{H}(j\omega)\} > 0 \forall \omega \in \mathbb{R}^+$.

This is a good loss function because

1. The dynamics of $\dot{V}$ are linear in g;
2. We can filter out the noise in $\dot{V}$ measurements through $\bar{H}$;
3. The loss function is quadratic in $(\dot{V} - \dot{\hat{V}})$, which (together with point 1.) allows us to use least squares methods to find $\hat{g}$.

From classical adaptive control theory [8, Theorem 2.5] it can then be shown that this loss function is minimized by the update rules

$$\begin{aligned} \dot{\hat{g}} &= \gamma P \psi^T (V - \hat{V}) \\ \dot{\psi} &= -\gamma\psi + \phi(V,\hat{w}) \\ \dot{P} &= \alpha P - P\psi^T \psi P \end{aligned} \tag{4.23}$$

where ψ is a matrix initialised so that $\psi(0) = 0$ and P a matrix so that $P(0) \succ 0$. These update rules are combined with the observer dynamics

$$\begin{aligned} \dot{\hat{V}} &= \phi(V,\hat{w})^T \hat{g} + I_{\text{app}} + \gamma(I + \psi^T P \psi)(V - \hat{V}) \\ \dot{\hat{w}} &= f(V,\hat{w}). \end{aligned} \tag{4.24}$$

Note that the dynamics of $\hat{V}$ depend on the measurements V, save for the feedback term in $\dot{\hat{V}}$.

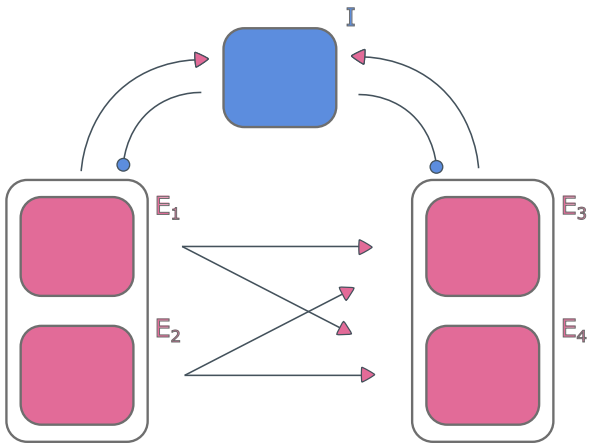


Fig. 4.9 A sketch of the PINGesque network which we use to test the teacher forcing method on.

A more extensive discussion on these observer methods is given in [35, 33, 34]. They all work on the premise that we know the inputs I_{app} and have measurements of voltage traces $V(t)$, and we are trying to estimate a model and parameters g that reproduce these measurements (where the degree to which we assume the model to be known can vary).

Our novel idea is that this can be reinterpreted as a training algorithm if $V(t)$ instead of a measurement is some desired set of voltage traces. Then we can use the observer methods, sketched above, to train a model to reproduce these desired outputs. In the rest of this chapter we shall refer to this method of training as *teacher forcing*, following the nomenclature for training methods where an instantaneous 'teacher' signal on the network's performance is given to the network in real-time [169, 17, 16] (in our case the $\gamma(I+\psi^T P\psi)(V-\hat{V})$ term in $\dot{\hat{V}}$).

To get this to work, we should have a method of synthesising voltage traces $V(t)$ which can then be used as a target for training. To demonstrate the validity of the method, we apply it to a toy example of a mini PING network. This network will consist of two excitatory input neurons, two excitatory output neurons, and one inhibitory neuron which is recurrently connected to all four other neurons.

To create our data, we cut the timeseries of a neuron simulation where the neuron spiked and a neuron that did not spike (only experienced the inhibitory wave coming from the I pool), and then concatenated these timeseries so the spikes and absence of spikes occured with a period T_{PING}. Because the timeseries did not neatly fit together, we used a linear interpolation to fill the gaps (see section 4.6.4 for details). This allowed us to generate artificial voltage traces based on binary words, where 1 represented a spike and 0 absence of a spike. This procedure is illustrated in Figure 4.10.

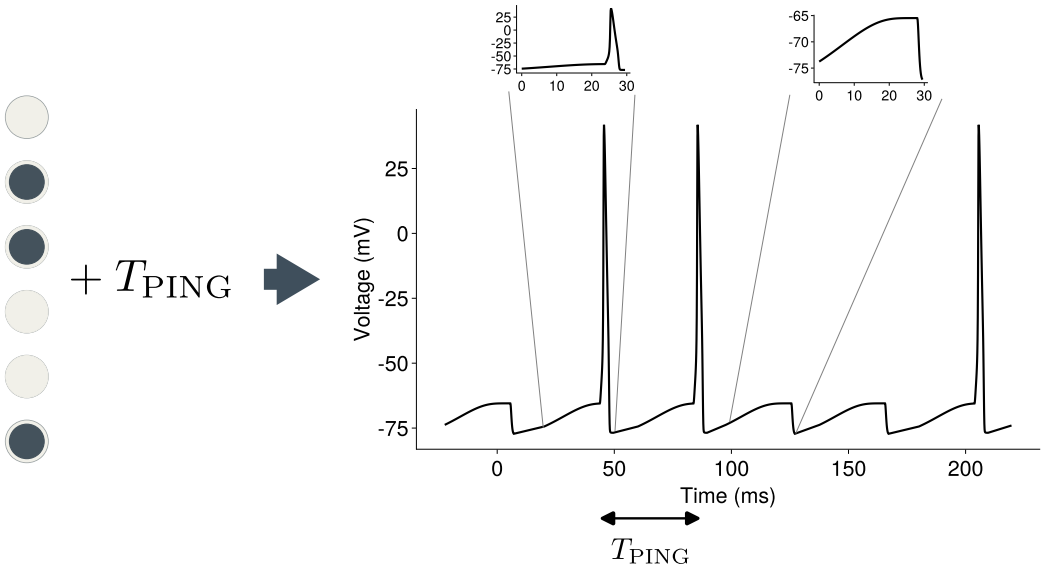


Fig. 4.10 Illustration of the construction of an artificial voltage trace based on a binary word an a timeperiod as inputs. The binary vector on the left is implemented as the target spiking output of a neuron through the artificial voltage shown on the right. The inlays at the top represent the 'spike' and 'no spike' time series respectively which were concatenated to form the overall target voltage trace.

Here, the target voltage traces used for training are obtained from a network with a PING architecture (Figure 4.9). The purpose of this is to ensure that the difference between the observer voltage trace and the training voltage trace is, insofar as is possible, only due to the wrong spiking behaviour. Even though the PING emerges naturally from the E–I motif, as we showed in chapter 3, and therefore does not need to be a training target, the feedback from the I population still changes the V trace. We want the error term $(\bar{H}\dot{V} - \bar{H}\dot{\hat{V}})^2$ to be unaffected by this, but only be affected by the erroneous presence or absence of spikes. Therefore, we find our training voltage traces in the presence of PING.

The goal of training a PING network is to have it perform as a BNN as outlined in section 4.1. We want to use the teacher forcing to implement this BNN. Our hypothesis is that if the PING neural circuit learns the same input-output spiking behaviour as its abstract BNN counterpart, it will be a PING implementation of that BNN computation. Should this be achievable, then the procedure to train a PING network would be as follows: firstly, we train a BNN, e.g., as we did in section 2.4.5. Secondly, we use this BNN, combined with our PING interpretation of a BNN (section 4.1) to generate the spiking data for each neuron in the PING neural circuit. Although the information we care about is the presence or absence of spikes, this target spiking data takes the form of voltage traces; the procedure to going from output spikes to fake voltage traces is outlined in Figure 4.10. Lastly, our hypothesis is that we can then use these voltage traces in our teacher forcing algorithm to train the SNN to perform the equivalent BNN computation. Due to time constraints we have not managed to confirm this hypothesis entirely. But, as an exploration of its validity, we constructed some toy problems to test whether, in principle, we can train an SNN using artificial input-output spiking data with teacher forcing.

With a method to generate artificial voltage time series in hand, we can let the method work on a few toy problems to demonstrate its promise. We choose simple toy problems where we know what the correct solution is. The only parameters we are trying to estimate are the synaptic weights; the rest of the model is considered to be known. For the precise equations which we implemented, see section 4.6.4.

We set two toy problems. First we randomly generated an input binary word, which was then realised by triggering spikes in E_1 and E_2. The one constraint was that both neurons should not be silent together, as that would interrupt the PING. So, for every input cycle, their spiking can either represent $[0,1]^T$, $[1,0]^T$ or $[1,1]^T$.

The first problem we tried is one where the objective is to have one of the E_3 and E_4 neurons spike continuously, whilst the other had to remain silent. We refer to this as the

on-off task. Of course, this task is accomplished by, for example, any weight matrix

$$W = \begin{bmatrix} a & 0 \\ 0 & 0 \end{bmatrix},$$

for sufficiently large positive a. The teacher forcing method is able to accomplish this task (Figure 4.11, bottom). Figure 4.11, top, shows the dynamics of the observer neurons (Equation 4.24) during training. The observer never converges exactly to the training data; this is unsurprising, since the training data was artificially created in a relatively crude manner. The training losses are shown in Figure 4.11, middle: it can be seen that the first spike gives a big loss, which then triggers a large correction in the synaptic weights. After this has occured there is a small error for each spike (which is conform with the discrepancy between observer and data in Figure 4.11, top). For this simple task, the training is good enough to correctly reproduce the spiking behaviour (Figure 4.11, bottom).

The second task we tried to get our mini network to perform was for each of the output neurons E_3, E_4 to mimic the spike train of one of the input neurons E_1, E_2. This task we dub the *diagonalisation* task, since this problem can easily be accomplished by an (anti-)diagonal weight matrix such as

$$W = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix},$$

for sufficiently large weights a and b. Results are shown in Figure 4.12, and are identical to those of the on-off task.

That we knew what the solutions of the training procedure looked like was important. This allowed us to verify that the tasks were indeed doable, and any failure in training was not due to the task being difficult or impossible. To verify that the training succeeded, we ran the network with the architecture from Figure 4.9 with the weight matrices that were the result of teacher forcing training. We dub these simulations the 'reproduction' simulations, shown in the bottom rows of Figure 4.11 and Figure 4.12. The output neurons follow the desired spiking trajectories, despite the observer never truly converging on a voltage trajectory. This lack of convergence (Figure 4.11 and Figure 4.12 top, middle) stems from the inherent discrepancy between realisable voltage traces and the voltage traces from the data generation. Nevertheless, the difference in spiketrains is substantial enough to allow for the correct behaviour (here interpreted as correct spiking output) to be learned.

Of course, the tasks here were very simple to achieve indeed; indeed, this is what allowed us to know a priori what the synaptic weight matrices broadly should look like. The toy problems above demonstrated that the artificial data generation works to learn spiking behaviour in principle, and from previous work [33] we know that, provided the data

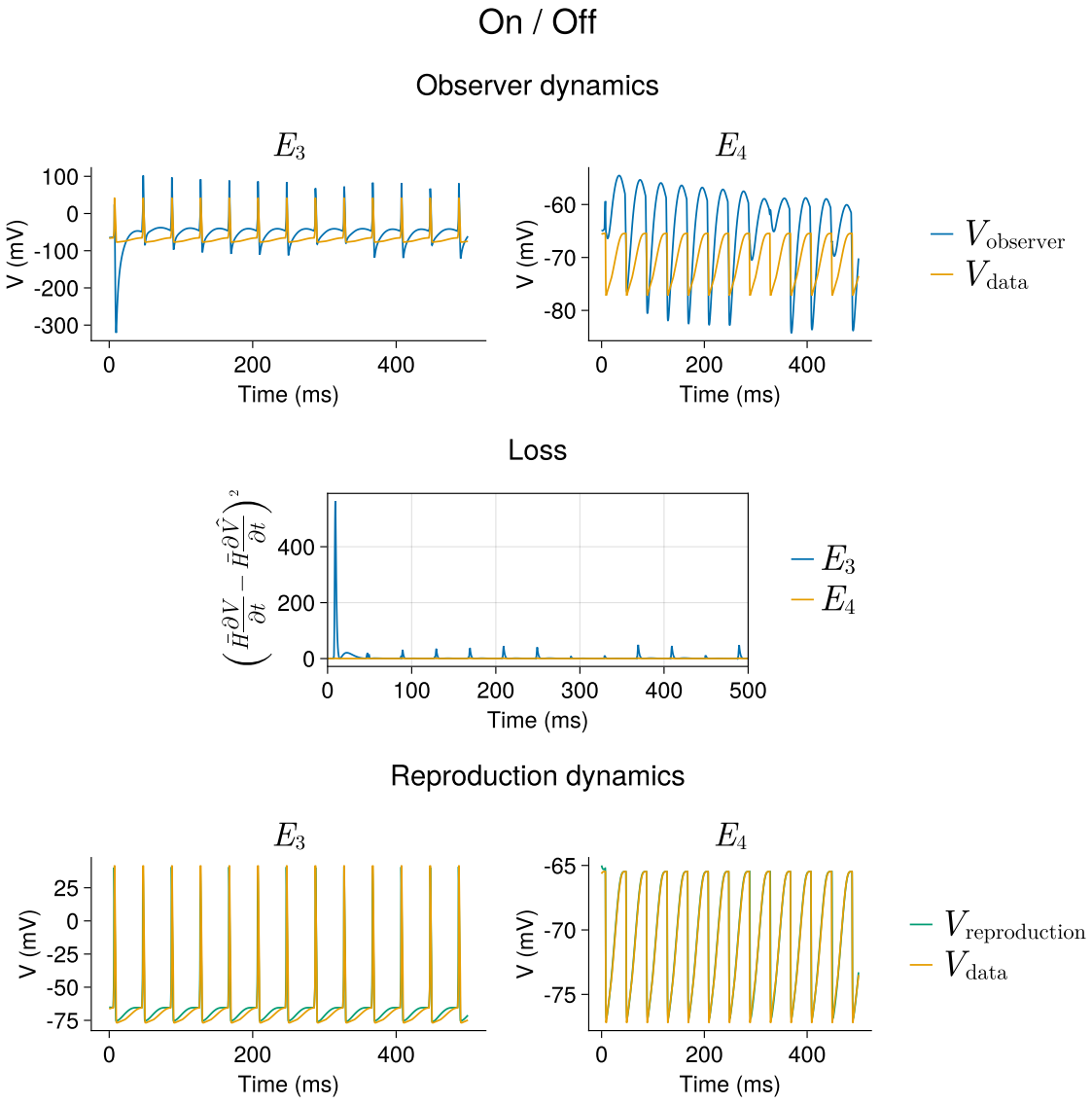


Fig. 4.11 The result of the teacher forcing training procedure on the on-off task. The voltage traces of the two observer neurons E_3 and E_4 during training (blue), together with their training data (orange), are shown on the top row, with the loss $(\bar{H}\dot{V} - \bar{H}\dot{\hat{V}})^2$ in the middle; the first wrong spike gives a large loss, with small loss for subsequent spikes. The voltage traces of the E_3, E_4 neurons in the reproduction (green) simulation on the bottom row, together with their training data (orange). The spikes in the data and the reproduced voltage are aligned, which shows that the correct spiking behaviour is learned.

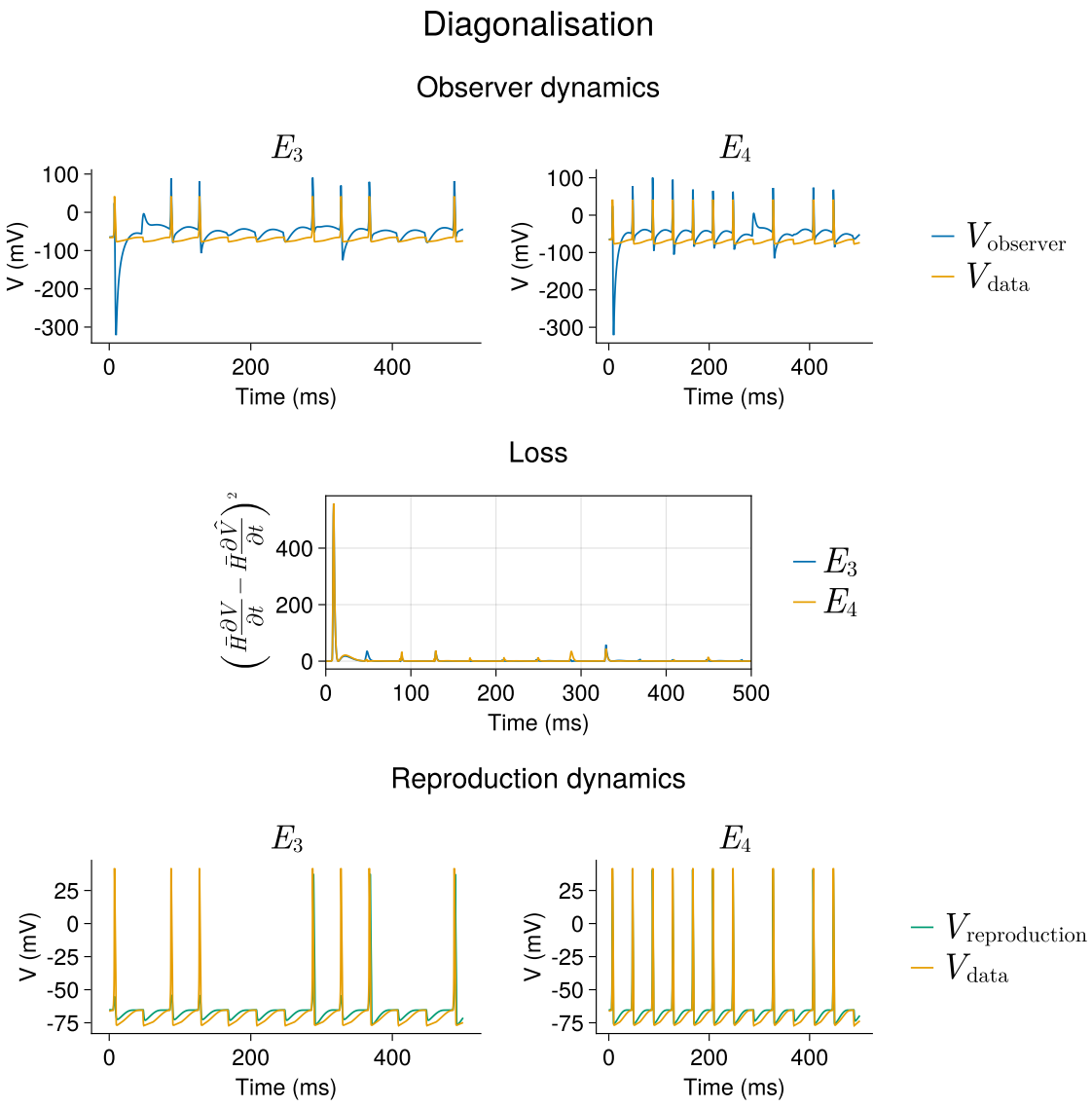


Fig. 4.12 The result of the teacher forcing training procedure on the diagonalisation task. The voltage traces of the two observer neurons E_3 and E_4 during training (blue), together with their training data (orange), are shown on the top row, with the loss $(\bar{H}\dot{V} - \bar{H}\dot{\hat{V}})^2$ in the middle; the first wrong spike gives a large loss, with small loss for subsequent spikes. The voltage traces of the E_3, E_4 neurons in the reproduction (green) simulation on the bottom row, together with their training data (orange). The spikes in the data and the reproduced voltage are aligned, which shows that the correct spiking behaviour is learned.

voltage trace is achievable by the model, it will converge on it. Therefore, the big question that remains is how big the sensitivity of teacher forcing is to the gap between the voltage trajectory of the observer and that of the generated data voltage, as can be seen, for example, in Figure 4.12, top. This is a question we aim to address in the near future.

4.4 Discussion

We introduced the idea that the activity of a feedforward PING network, consisting of several layers of E pyramidal neurons all recurrently coupled to a population of I inhibitory interneurons, can be thought of as performing a binary computation. The tight synchrony of the PING in combination with the shortness of the timescales of the dynamics of the single neurons mean that neurons cannot integrate or fire more than 1 spike each. The BNN interpretation as an abstract computational notion is therefore a natural one, as it is a flexible and straightforward way to compute in a regime where firing rate models would struggle to work within the confines of a tight energy budget, which we outlined in chapter 2.

We briefly demonstrated that neuron participation can vary without damaging the overall rhythm. This is important, as it allows for flexibility in the computation of the circuit, whilst the mechanism that facilitates the computation is robust. The information transfer through PING circuits is good, provided the time gap between inputs exceeds the period of the oscillation.

Our results show that training spiking networks where neurons have a threshold which is shaped by a mixture of positive and negative feedback is not easy. The positive and negative feedback lead to amplification and suppression of the output respectively, in different voltage ranges. We demonstrated that this causes issues by comparing the training of LIF-based networks using surrogate gradient methods with the training of neurons possessing more complicated dynamics. The feedback in the neurons caused the training to fail, where it worked easily with the LIF networks. The question of how to get around these difficulties with surrogate gradient methods is an interesting question, to which we will return in future work.

One way of remedying these difficulties is by switching training paradigms. We introduced a teacher forcing method, which works by optimising conductances of neurons to cause them to follow a prescribed voltage trajectory. We demonstrated that this worked for a toy example; namely training a PING network with two input neurons and two output neurons. We found the ease with which the method was implemented encouraging. Nevertheless, how well it generalises, and its drawbacks, are to be examined in future work.

4.5 Methods

4.6 PING BNN

The dynamics of all the neurons are identical to those in section 3.4. For Figure 4.2, the values of μ looped over are between 11 and 14, and for σ between 0.2 and 0.3, both in five equidistant steps.

For Figure 4.3, to measure the mutual information between the sets of input and output vectors, the CasaulityToolsjl package [61], using the `mutualinfo(Contingency(), X, Y)` function, with X and Y being the matrices of inputs and outputs respectively.

4.6.1 Feedback sensitivities

The sensitivities of the Hodgkin-Huxley neuron (Figure 4.5) were computed numerically, using Zygote.jl [81] for the automatic differentiation.

To compute the sensitivity of the MQIF model (Equation 4.16), we took $I_m(V,V_s) = -g_f\left(V-V^0\right)^2 + g_s\left(V_s-V_s^0\right)^2$ and computed

$$\begin{aligned} H(s,V) &= \left.\frac{\partial I_m(V,V_s)}{\partial V}\right|_{(V,V)} + \left.\frac{\partial I_m(V,V_s)}{\partial V_s}\right|_{(V,V)} \frac{V_{s,\infty}}{\partial V}\frac{1}{\tau_s s+1} \\ &= -2g_f(V-V^0) + \frac{2g_s(V-V_s^0)}{\tau_s s+1}. \end{aligned} \tag{4.25}$$

Then,

$$L(s,V) = \frac{1}{Cs}\left[-2g_f(V-V^0) + \frac{2g_s(V-V_s^0)}{\tau_s s+1}\right]. \tag{4.26}$$

To maximise $\max_{\omega\in\mathbb{R}^+}|1/(1+L(i\omega,V))|$ we first take the square, and then take the derivative to find the ω that maximises this expression. (We can of course do this, because the squared expression will be maximised for the same ω as the expression with the absolute value.) Doing this for the MQIF yields

$$\begin{aligned} &\frac{\partial}{\partial\omega}\left(\frac{1}{1+L(i\omega,V))}\right)^2 = \\ &\frac{2i}{\left(1+\frac{2(V-V_{s0})g_s}{C\omega}\cdot\frac{1-i\omega\tau_s}{1+(\omega\tau_s)^2}-\frac{2(V-V_0)g_f}{C\omega}\right)^3}\cdot \\ &\left(\frac{2(V-V_{s0})g_s\tau_s+2(V-V_{s0})g_s}{C\omega}\cdot\frac{(1-i\omega\tau_s)^2}{(1+(\omega\tau_s)^2)^2}-\frac{2(V-V_0)g_f}{C\omega}\right) \end{aligned} \tag{4.27}$$

to solve

$$\frac{\partial}{\partial \omega} \max_{\omega \in \mathbb{R}^+} \left(\frac{1}{1 + L(i\omega, V))} \right)^2 = 0, \tag{4.28}$$

we obtain a fourth-degree polynomial in ω. This we solved numerically using a root-finding algorithm.

The sensitivities of the LIF model are simple to calculate. If the LIF model has leak conductance g_{leak} then

$$C\dot{V} = -I_m(V) = -g_{\text{leak}}\,(V - V_{\text{rest}})\,, \tag{4.29}$$

and therefore

$$\begin{aligned} H(s,V) &= \frac{dI_m}{dV} = g_{\text{leak}} \\ \therefore L(s,V) &= \frac{g_{\text{leak}}}{Cs} \\ \implies \left| \frac{1}{1 + L(i\omega, V))} \right| &= \text{hypot}\left(\frac{1}{1 + \left(\frac{g_l}{C\omega}\right)^2}, \frac{\frac{g_l}{C\omega}}{1 + \left(\frac{g_l}{C\omega}\right)^2} \right), \end{aligned} \tag{4.30}$$

and we can see by inspection that this function has a supremum of 1 in the limit $\omega \to \infty$, but the important point is that $L(s,V) = L(s)$, i.e., it is independent of voltage.

4.6.2 Surrogate gradients

The synthetic dataset which we used in the surrogate gradient was taken from [149]. We generated 100 input spiketrains that, on average, contain 5 spikes per second. A stepsize of $\Delta t = 1\,\text{ms}$ was used, with a batchsize of 256, for 200 timesteps. Specifically, we generated an array of dimension $(256, 200, 100)$, and made every element 1 with probability $p = 5\Delta t$.

The target labels that the network had to guess was an array of 100 bits, where every bit was set equal to 1 with probability $p = 0.5$, so that every set of input spike trains was randomly assigned to one of two classes. The task of the SNN was to reproduce this classification. Training occured over 1000 epochs.

4.6.3 MQIF model

The values used during the simulations with MQIF neurons are given in Table 4.1. We chose these parameters so that the MQIF model behaved as a Type II neuron (see also [159, Figure 12A]) without bursting.

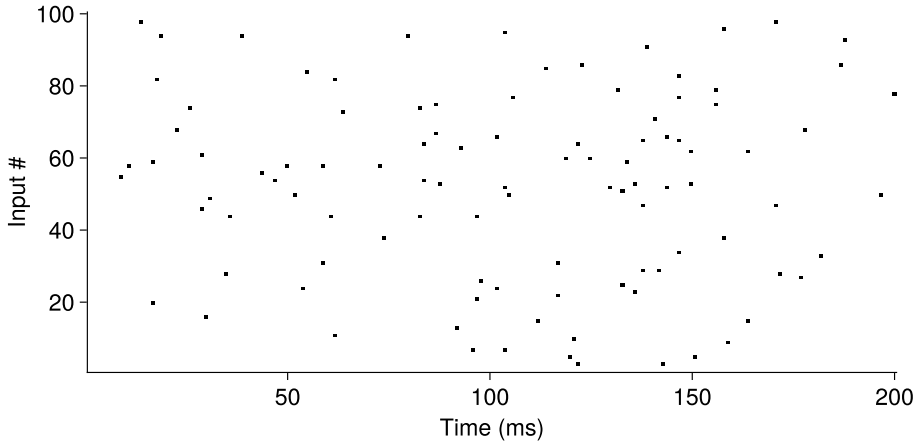


Fig. 4.13 An example of the collection of spiketrains (one out of 256) used during the training of the LIF and MQIF based SNNs.

Table 4.1 Standard values of parameters used in our MQIF model simulations.

C	g_f	g_s	V^0	V_s^0	V_r	$V_{s,r}$
$1\,\mu\mathrm{F/cm^2}$	$1\,\mathrm{mS/cm^2}$	$0.5\,\mathrm{mS/cm^2}$	$-40\,\mathrm{mV}$	$-41\,\mathrm{mV}$	$-40\,\mathrm{mV}$	$-35\,\mathrm{mV}$

The resting potential of the MQIF model can be computed analytically and is

$$V_{\text{rest}} = \frac{V^0 + (g_s/g_f)V_s^0}{1 + g_s/g_f} + \frac{1}{3}. \tag{4.31}$$

To use the exponential Euler we first must separate the ODE of the MQIF model into a linear and nonlinear part:

$$\begin{aligned} C\dot{V} &= g_f\left(V - V^0\right)^2 - g_s\left(V_s - V_s^0\right)^2 + I \\ &= g_f\left(V^2 - 2VV^0 + V^{0^2}\right) - g_s\left(V_s - V_s^0\right)^2 + I \\ &= \underbrace{V\left(-2g_fV^0\right)}_{\text{linear part}} + \underbrace{\left[g_f\left(V^2 + V^{0^2}\right) - g_s\left(V_s - V_s^0\right)^2 + I\right]}_{\text{nonlinear part}}. \end{aligned} \tag{4.32}$$

We can write this as

$$\tau_V\dot{V} = V_\infty - V \tag{4.33}$$

by defining

$$\begin{aligned} \tau_V &:= \frac{C}{2\bar{g}_f V^0} \\ V_\infty &:= \frac{\bar{g}_f \left(V^2 + V^{0^2}\right) - g_s \left(V_s - V_s^0\right)^2 + I}{2\bar{g}_f V^0}, \end{aligned} \tag{4.34}$$

We can define, $h(V[n]) := V_\infty(V[n], V_s[n]) + [V[n] - V_\infty(V[n], V_s[n])]\, e^{-\Delta t/\tau_V}$ then the exponential Euler update step becomes

$$\begin{aligned} V_{n+1} &= h(V_n) + H(V_n - V_{\max})[-h(V_n) + V_r] \\ &= h(V_n)[1 - H(V[n] - V_{\max})] + \underbrace{H(V_n - V_{\max}) V_r}_{\text{reset term}}. \end{aligned} \tag{4.35}$$

For $\dot{V}_s$ we define $k(V_s[n]) := V[n] + (V_s[n] - V[n])\, e^{-\Delta t/\tau_i}$. Then

$$V_s[n] = k(V_s[n]) + \underbrace{H(V[n] - V_{\max})[-k(V_s[n]) + V_{s,r}]}_{\text{reset term}}. \tag{4.36}$$

To make the connection with the LIF model more explicit, we rescaled the MQIF voltage so that its resting potential lied at 0 and its threshold at 1. To achieve this we did

$$\tilde{V} = (V - V_{\text{rest}})/V_{\max}, \tag{4.37}$$

and then multiplied $\tilde{C} = V_{\max} C$ so that $\tilde{C}\dot{\tilde{V}} = C\dot{V}$.

During the MQIF surrogate gradient training, we rescaled the weights so that the average number of spikes in the MQIF network matched the average number of spikes in the LIF network. Suppose neurons in the hidden layer receive inputs $h_1[n]$ from the first layer, and we define $\beta_V := e^{\Delta t/\tau_V}$. Then the update step becomes

$$\begin{aligned} s[n+1] &= \beta_{\text{syn}} s[n] + h_1[n] \\ V[n+1] &= (V_\infty(V[n], s[n]) + \beta_V \{V[n] - V_\infty(V[n], s[n])\})[1 - H(V[n] - V_{\max})] \\ &\quad + H(V[n] - V_{\max}) V_r \\ V_\infty &:= \frac{\bar{g}_f \left(V^2 + V^{0^2}\right) - g_s \left(V_s - V_s^0\right)^2 + s[n]}{2 g_f V^0}. \end{aligned} \tag{4.38}$$

We see that the V_∞ dependence on the synaptic input is

$$V_\infty \sim s[n] / \left(2 g_f V^0\right), \tag{4.39}$$

and therefore that

$$\begin{aligned} V[n+1] &\sim \frac{s[n]}{2\bar{g}_f V^0} + \left(V[n] - \frac{s[n]}{2 g_f V^0}\right) \beta_V \\ &= \beta_V V[n] + s[n] \left(\frac{1-\beta_V}{2\bar{g} V^0}\right). \end{aligned} \tag{4.40}$$

Compare this with the LIF update step

$$V[n+1] = \beta V[n] + s[n], \tag{4.41}$$

and we see that rescaling all the weights as

$$w \mapsto \left(\frac{1-\beta_V}{2 g V^0}\right)^{-1} w \tag{4.42}$$

should recover the average activity in the MQIF network, which we verified empirically.

4.6.4 Teacher forcing

We simulated the network in Figure 4.9 using the same methods as in section 3.2.

To create our voltage data to use on the teacher forcing, we simulated the network from Figure 4.9 with the weight matrix

$$W = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

This ensured that the E_3 neuron was seeing both feedforward excitation and inhibition, whilst E_4 only saw inhibition. From this simulation data, we cut the timeseries of a spike of E_3 and an inhibitory transient in E_4 for a length of 30 ms.

We implement distributed single neuron observers [33, Remark 8], so that every neuron in the network we want to train is replaced by an observer that tries to estimate its incoming

synaptic weights. If the neuron dynamics are represented by

$$\dot{v} = \phi_{\text{internal}}(v,x)^T \bar{g} + \phi_{\text{synaptic}}(v,s)^T \bar{w} + I_{\text{app}}(t), \text{ voltage dynamics}$$

$$\phi_{\text{internal}} = \begin{bmatrix} \vdots \\ x_{i1}^p x_{i2}^q \left(E_{x_i} - v\right) \\ \vdots \end{bmatrix} \text{ internal currents (without maximal conductances),} \tag{4.43}$$

$$\phi_{\text{synaptic}} = \begin{bmatrix} \vdots \\ s_{ij}(v_i)\left(E_{s_{ij}} - v\right) \\ \vdots \end{bmatrix} \text{ synaptic currents (without synaptic weights),}$$

then the dynamics of each observer consist of

$$\text{Observer dynamics} \begin{cases} \dot{\widehat{V}} & = \phi_{\text{internal}}(V,\widehat{x})^T \bar{g} + \phi_{\text{synaptic}}(V,\widehat{s})^T \widehat{w} + I_{\text{app}}(t) \\ & \quad + \gamma(I + \psi^T P \psi)(V - \widehat{V}) \\ \dot{\widehat{x}} & = f(\widehat{x},V) \\ \dot{\widehat{s}} & = h(\widehat{s}, V_{\text{presynaptic}}) \end{cases}, \tag{4.44}$$

$$\text{Update rule} \begin{cases} \dot{\widehat{w}} & = \gamma P \psi^T (V - \widehat{V}) \\ \dot{\psi} & = -\gamma\psi + \phi_{\text{synaptic}}(V,s) \\ \dot{P} & = \alpha P - P\psi^T \psi P \end{cases}, \tag{4.45}$$

where we have now written x for the gating variables to avoid confusion with the synaptic weights w. $V_{\text{presynaptic}}$ are the voltages of the presynaptic neurons and s are their associated postsynaptic conductances. Note that the only parameters affected by the update rule are these synaptic weights w.

Chapter 5

Discussion

The question this thesis has sought to partially address, is that of how the brain could compute with action potentials, also known as spikes. To start with answering this question, one has to consider the biological constraints that neural circuits operate in. A few of these constraints are important for our considerations.

Perhaps the most salient feature of spiking computations is that sometimes they can be carried out rapidly. For illustration, we recall the calculation done back in the 80s by Simon Thorpe [154]. He examined data from experiments done with macaque monkeys. In these experiments, experimenters found neurons that were responsive to facial features which responded within roughly 100 ms of the stimulus presentation. Thorpe computed that the shortest possible anatomical pathway to these neurons still crossed ten synaptic stages. This leaves about 10 ms per neuron to receive spikes and make a firing decision for itself. Even with the firing rates in the very highest ranges observed in the brain, this would leave about one spike per neuron to fire. This biological fact, and others (see chapter 1), imply that there are times when a biological network must operate in a regime where neurons can only fire one spike each.

Such a regime is attractive for several reasons. One big advantage is speed: the fewer spikes each neuron has to fire, the quicker the circuit can perform its tasks. It also saves energy. Action potentials are very costly energetically speaking [9], so the more of them are required, the more demand is placed on the energy budget of the brain. Therefore, there is evolutionary pressure to move towards an action potential efficient regime where neurons fire less often. This is in line with numerous other features present in the nervous system which serve to optimise efficiency and conserve energy [150]. And from a scientific point of view, it seems more satisfactory to assume that neurons fire their spikes in a reasonably precise fashion for good reason, as they produce reliable spikes, or sequences of spikes, in

experimental settings [104, 12, 2]. It appears a strong assumption that this would not be exploited by the brain, but a mere accident.

5.1 Recapitulation

With these biological constraints in mind we want to build a spiking neural network which uses biologically plausible components in order to work. In chapter 2, we explored why this is not a trivial thing to do. Naive approaches to build SNNs struggle when timing jitter is introduced. Circuit function becomes unreliable, and can only be rescued by cranking up the firing rates of the neurons involved, which runs counter to the constraints above. Such timing jitter is unavoidable due to conduction delays and noise in communication between circuits in the brain [53]. We showed how, paradoxically, the slow dynamics of dendritic action currents can make rapid spiking computations robust. We showed that the slow time constants, provided by NMDA depolarization events in dendritic branches, allowed neurons to widen their integration time window, thereby increasing their tolerance to jitter in spike timings.

This part described an intra-neuronal mechanism for SNNs to deal with spike timing jitter. A conspicuous omission is the role of inhibition, which is known to play an important role in determining when and if neurons can fire in a network. Inhibition came to the foreground in chapter 3, where we showed that the interplay of exhibition and inhibition on the network level can promote synchrony and thereby reduce spike timing jitter. We considered PING rhythms, and demonstrated that these oscillations appear easily when building networks of biophysically plausible neurons. We showed that these PING rhythms remain robust when the network is faced with disturbances such as exogenous inputs or parameter uncertainty.

In order to analyse these network-wide effects, we introduced a novel tool: the population voltage clamp. This method for analysis takes the analogy between feedback loops on the network level and feedback loops on the cell level (i.e. ion channels) at face value, and uses a network wide voltage clamp to extract the feedback a recurrent pathway provides on a particular type of neurons in a network in several timescales. Applying it to the PING rhythm, we found that there is slow positive feedback on the network level underpinning it, as a result of the E–I feedback. This explains why the rhythm does not always exist when using non-biophysical neuron models in the network, such as LIF neurons: the lack of internal dynamics on slower timescales precludes slow positive feedback, and the presence of the oscillation is brittle.

PING is a good candidate to build spiking circuits with. We demonstrated that it is easy to obtain, and it is robust. It also covers the timescale of the computation we want to

implement; with a frequency band of 30 Hz to about 90 Hz it is suitable for computations with time windows of about 10 ms. In chapter 4, we showed that PING rhythms maintain good coherence for a variable number of neurons participating, provided that a sufficient number of excitatory neurons is active during a cycle to excite the pool of inhibitory neurons. This means that it can support flexible functions where the neurons can drop in and out of the computations as needed. Furthermore, we showed that the output of PING networks reflects the inputs, as measured by their mutual information, and that this vanishes as the synchrony disappears.

We closed off in chapter 4 by outlining the difficulties of training SNNs using gradient-based methods in a naive way. Biophysical spiking systems have a threshold which is shaped by the nesting of fast positive feedback amidst slower negative feedback [50, 141]. In such systems, by definition, the threshold is a narrow region in the voltage where the output becomes hypersensitive to variations in the input: when just below threshold, a miniscule increase in excitation leads to a spike. This hypersensitivity means that gradients of the input-output mapping of neurons will explode in that region. Conversely, when the voltage is not near threshold, the effect of inputs is attenuated, and the input-output gradients vanish.

The existence of this effect was confirmed by comparing the training of standard LIF neurons, which are the current standard in spiking neural networks, with the training of a more complicated MQIF neural network. To circumvent these difficulties, we proposed an alternative training method: that of teacher forcing. In this method, tools from nonlinear system identification are used, to adapt maximal conductance values so that neurons follow a prescribed voltage trajectory. We demonstrated that this works in a toy example network performing some simple tasks.

5.2 Future work

The main objective that remains is to construct a neural circuit which exhibits PING oscillations and is trained as a binary neural network on a task of our choosing. The teacher forcing method discussed in section 4.3.3 provides one possible avenue to achieve this goal.

Before we are in a position to train BNN-esque PING spiking circuits, we must understand better the sensitivities of teacher forcing to noise in the target voltage trajectories. From previous work [35, 33] it is known that, given sufficient data and some lenient conditions on the data being sufficiently rich to specify the required dynamics, the model behaviour will converge on the desired behaviour. But this result holds when it is known that the model can produce the voltage trajectories that it is trained on. In our case the voltage trajectories are ‘fake’; they are individual voltage traces glued together to specify the desired spike

timings of the neurons in the network. Due to successes of applying the system identification algorithm on real neurons, which includes the noise from measurements and the filtering of spikes being propagated from the soma to the location of the electrode, the ease with which the toy networks were trained (simple though they were), and the freedom we have in both specifying the surrogate voltage traces and the filters in the equations, we are optimistic that this method will work.

An important open question is that of the robustness of binary computations in a PING network. One way in which the computation might fail is by having too many successive layers of excitatory neurons, all connected to a single inhibitory pool. It could take too long for the signal to finish propagating through the network, and the wave of inhibition could come in before the SNN has settled on an answer. Different architectures can be explored here; for example, one can play with having each excitatory layer having its own private inhibitory population. In such networks the inhibition would travel through space as well as time, and the maximum depth could be much bigger.

In the same context the connection between the active dendrites and the PING networks can be made. Our hypothesis is that the integrate-and-hold mechanism can serve to permit information being preserved throughout the phase of neural oscillations present in the network, in addition to providing the increased computational robustness we discussed in chapter 2. Interesting avenues of research here can include having multiple neuronal PING circuits, in different phases of their computation, having to share their results with each other. The active dendrites could serve to hold the information for a circuit until it finishes its PING cycle and is ready to compute.

In section 3.2.1, we briefly touched upon the theoretical attractiveness of the voltage clamp-based E–I mean field reduction of PING. This mean-field reduction bridges the gap between conductances of ion channels and the network-wide behaviour of the circuit. This opens up new avenues for exploring the connections between the two. One area where this might be exploited is in the tuning of the PING rhythm, by making predictions on how some changes in conductances impacts the rhythm. A hypothesis we are keen to explore is that some neuromodulators can decrease the PING frequency, thereby changing the computation. With longer time periods involved, neurons get the chance to fire and integrate multiple spikes instead of at most one in the fastest limit, which is the BNN regime we have discussed hitherto. Instead of a binary limit, they now operate in a more ‘rate-like’ regime, where more uncertainty can be taken into account. Some neuromodulators, such as norepinephrine, might increase the rythm’s frequency, thereby speeding up circuit’s computation, which puts the organism is a ‘fight or flight’ mode; a brain state where speed is favoured at the expense of accuracy. Conversely, a neuromodulator like serotonin might have the opposite effect, namely

promoting precision of the neural computation at the expense of speed. This hypothesis is motivated by Simon Thorpe's experiments [154] that we have mentioned already, who showed that we can make decisions at the fastest time possible with one spike per neuron He also showed that given more time, we make better decisions. This is the neuronal analogy of being in a situation where one is forced to answer a question instantaneously versus one where time is given for deliberation; in the latter, they are slower, but the computation is performed more accurately.

In this thesis we have only disussed the possibility of feedforward computations. Once a PING binary computation is achieved, we want to examine the possibility of extending the idea to recurrent neural networks. One possibility to train recurrent neural networks is by unrolling the network to represent it as a multi-layer feedforward network (with an unbounded number of layers) and training this unrolled model. We hypothesise that this method will apply to the PING networks also, thereby allowing us to construct recurrent versions of it. A known difficulty in doing so is the occurrence of vanishing and exploding gradients [121], even with traditional ANN architectures. We hypothesise that teacher-forcing will alleviate these issues, from successes in applying it to small recurrent physiological networks [32].

Finally, we would like to explore the possibility of using gradient-based methods in spiking systems in the future. In section 4.3.1, we examined some of the difficulties that such an approach can encounter. It would be interesting to investigate how these challenges can be remedied. Moreover, we want to investigate the possibility of ruling certain learning rules for the brain in or out.

5.3 Final remarks

This thesis sought to help fill the gap in neuroscience between the elaborate biophysical models of single neurons or small neuronal circuits, with which it is difficult to build anything that actually does a task, and the very useful, but abstract and biologically implausible, artificial neural networks used today. We hope that the ideas and methods presented here prove useful in achieving this objective.

References

[1] M. Abeles. *Corticonics: Neural Circuits of the Cerebral Cortex*. Cambridge University Press, Feb. 22, 1991. 298 pp. ISBN: 978-0-521-37617-4. Google Books: v46SDOLJrLcC.

[2] M. Abeles et al. "Spatiotemporal Firing Patterns in the Frontal Cortex of Behaving Monkeys". In: *Journal of Neurophysiology* 70.4 (Oct. 1993), pp. 1629–1638. ISSN: 0022-3077. DOI: 10.1152/jn.1993.70.4.1629. URL: https://journals.physiology.org/doi/abs/10.1152/jn.1993.70.4.1629 (visited on 11/20/2023).

[3] Corey D. Acker and Srdjan D. Antic. "Quantitative Assessment of the Distributions of Membrane Conductances Involved in Action Potential Backpropagation Along Basal Dendrites". In: *Journal of Neurophysiology* 101.3 (Mar. 2009), pp. 1524–1541. ISSN: 0022-3077. DOI: 10.1152/jn.00651.2007. URL: https://journals.physiology.org/doi/full/10.1152/jn.00651.2007 (visited on 03/14/2022).

[4] E. D. Adrian. *The Mechanism of Nervous Action: Electrical Studies of the Neurone*. University of Pennsylvania Press, Nov. 11, 2016. 116 pp. ISBN: 978-1-5128-0979-4. Google Books: WVArEAAAQBAJ.

[5] E. D. Adrian and Yngve Zotterman. "The Impulses Produced by Sensory Nerve-Endings". In: *The Journal of Physiology* 61.2 (1926), pp. 151–171. ISSN: 1469-7793. DOI: 10.1113/jphysiol.1926.sp002281. URL: https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1926.sp002281 (visited on 08/17/2019).

[6] Srdjan D. Antic et al. "The Decade of the Dendritic NMDA Spike". In: *Journal of Neuroscience Research* 88.14 (2010), pp. 2991–3001. ISSN: 1097-4547. DOI: 10.1002/jnr.22444. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jnr.22444 (visited on 03/08/2022).

[7] Karl J. Åström and Richard M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton: Princeton University Press, 2008. 396 pp. ISBN: 978-0-691-13576-2.

[8] Karl J. Åström and Björn Wittenmark. *Adaptive Control*. 2nd ed., Dover ed. Dover Books on Engineering. Mineola, N.Y: Dover Publications, 2008. 573 pp. ISBN: 978-0-486-46278-3.

[9] David Attwell and Simon B. Laughlin. "An Energy Budget for Signaling in the Grey Matter of the Brain". In: *Journal of Cerebral Blood Flow & Metabolism* 21.10 (Oct. 1, 2001), pp. 1133–1145. ISSN: 0271-678X. DOI: 10.1097/00004647-200110000-00001. URL: https://doi.org/10.1097/00004647-200110000-00001 (visited on 08/13/2019).

[10] Rony Azouz and Charles M. Gray. "Adaptive Coincidence Detection and Dynamic Gain Control in Visual Cortical Neurons in Vivo". In: *Neuron* 37.3 (2003), pp. 513–523. URL: https://www.cell.com/neuron/pdf/S0896-6273(02)01186-8.pdf (visited on 11/10/2023).

[11] Wyeth Bair and Lawrence P. O'keefe. "The Influence of Fixational Eye Movements on the Response of Neurons in Area MT of the Macaque". In: *Visual Neuroscience* 15.4 (Apr. 1998), pp. 779–786. ISSN: 1469-8714, 0952-5238. DOI: 10.1017/S0952523898154160. URL: https://www.cambridge.org/core/journals/visual-neuroscience/article/influence-of-fixational-eye-movements-on-the-response-of-neurons-in-area-mt-of-the-macaque/D409144D59369B583A1AB0417123450F (visited on 08/20/2019).

[12] H B Barlow. "Single Units and Sensation: A Neuron Doctrine for Perceptual Psychology?" In: *Perception* 1.4 (Dec. 1, 1972), pp. 371–394. ISSN: 0301-0066. DOI: 10.1068/p010371. URL: https://doi.org/10.1068/p010371 (visited on 11/13/2023).

[13] H. B. Barlow et al. "Human Contrast Discrimination and the Threshold of Cortical Neurons". In: *JOSA A* 4.12 (Dec. 1, 1987), pp. 2366–2370. ISSN: 1520-8532. DOI: 10.1364/JOSAA.4.002366. URL: https://opg.optica.org/josaa/abstract.cfm?uri=josaa-4-12-2366 (visited on 11/19/2023).

[14] Edward L. Bartlett and Xiaoqin Wang. "Correlation of Neural Response Properties with Auditory Thalamus Subdivisions in the Awake Marmoset". In: *Journal of Neurophysiology* 105.6 (June 2011), pp. 2647–2667. ISSN: 0022-3077. DOI: 10.1152/jn.00238.2010. URL: https://journals.physiology.org/doi/full/10.1152/jn.00238.2010 (visited on 11/20/2023).

[15] Ramon Bartolo et al. "Information-Limiting Correlations in Large Neural Populations". In: *Journal of Neuroscience* 40.8 (Feb. 19, 2020), pp. 1668–1678. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.2072-19.2019. pmid: 31941667. URL: https://www.jneurosci.org/content/40/8/1668 (visited on 08/04/2022).

[16] George Bekey and Kenneth Y. Goldberg. *Neural Networks in Robotics*. Springer Science & Business Media, Nov. 30, 1992. 582 pp. ISBN: 978-0-7923-9268-2. Google Books: d9HpQBWRcXoC.

[17] Nikzad Benny Toomarian and Jacob Barhen. "Learning a Trajectory Using Adjoint Functions and Teacher Forcing". In: *Neural Networks* 5.3 (Jan. 1, 1992), pp. 473–484. ISSN: 0893-6080. DOI: 10.1016/0893-6080(92)90009-8. URL: https://www.sciencedirect.com/science/article/pii/0893608092900098 (visited on 11/01/2023).

[18] O Bernander et al. “Synaptic Background Activity Influences Spatiotemporal Integration in Single Pyramidal Cells.” In: *Proceedings of the National Academy of Sciences* 88.24 (Dec. 15, 1991), pp. 11569–11573. DOI: 10.1073/pnas.88.24.11569. URL: https://www.pnas.org/doi/abs/10.1073/pnas.88.24.11569 (visited on 03/10/2022).

[19] Christoph Börgers, Steven Epstein, and Nancy J. Kopell. “Background Gamma Rhythmicity and Attention in Cortical Local Circuits: A Computational Study”. In: *Proceedings of the National Academy of Sciences* 102.19 (May 10, 2005), pp. 7002–7007. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0502366102. pmid: 15870189. URL: https://www.pnas.org/content/102/19/7002 (visited on 08/05/2019).

[20] Christoph Börgers and Nancy Kopell. “Synchronization in Networks of Excitatory and Inhibitory Neurons with Sparse, Random Connectivity”. In: *Neural Computation* 15.3 (Mar. 2003), pp. 509–538. ISSN: 0899-7667. DOI: 10.1162/089976603321192059.

[21] S. Boyd and L. Chua. “Fading Memory and the Problem of Approximating Nonlinear Operators with Volterra Series”. In: *IEEE Transactions on Circuits and Systems* 32.11 (Nov. 1985), pp. 1150–1161. ISSN: 0098-4094. DOI: 10.1109/TCS.1985.1085649. URL: http://ieeexplore.ieee.org/document/1085649/ (visited on 09/26/2023).

[22] A. Bragin et al. “Gamma (40-100 Hz) Oscillation in the Hippocampus of the Behaving Rat”. In: *Journal of Neuroscience* 15.1 (Jan. 1, 1995), pp. 47–60. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.15-01-00047.1995. pmid: 7823151. URL: https://www.jneurosci.org/content/15/1/47 (visited on 08/24/2023).

[23] Valentino Braitenberg and Almut Schüz. *Cortex: Statistics and Geometry of Neuronal Connectivity*. Springer Science & Business Media, Mar. 14, 2013. 243 pp. ISBN: 978-3-662-03733-1. Google Books: yGvvCAAAQBAJ.

[24] Tiago Branco, Beverley A. Clark, and Michael Häusser. “Dendritic Discrimination of Temporal Input Sequences in Cortical Neurons”. In: *Science* 329.5999 (Sept. 24, 2010), pp. 1671–1675. DOI: 10.1126/science.1189664. URL: https://www.science.org/doi/full/10.1126/science.1189664 (visited on 03/08/2022).

[25] Tiago Branco and Michael Häusser. “Synaptic Integration Gradients in Single Cortical Pyramidal Cell Dendrites”. In: *Neuron* 69.5 (Mar. 10, 2011), pp. 885–892. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2011.02.006. URL: https://www.sciencedirect.com/science/article/pii/S0896627311001036 (visited on 08/05/2022).

[26] Tiago Branco and Michael Häusser. “The Single Dendritic Branch as a Fundamental Functional Unit in the Nervous System”. In: *Current Opinion in Neurobiology*. Signalling Mechanisms 20.4 (Aug. 1, 2010), pp. 494–502. ISSN: 0959-4388. DOI: 10.1016/j.conb.2010.07.009. URL: https://www.sciencedirect.com/science/article/pii/S0959438810001170 (visited on 02/14/2023).

[27] Mary a. B. Brazier. “THE DEVELOPMENT OF CONCEPTS RELATING TO THE ELECTRICAL ACTIVITY OF THE BRAIN”. In: *The Journal of Nervous*

and Mental Disease 126.4 (Apr. 1958), p. 303. ISSN: 0022-3018. URL: https://journals.lww.com/jonmd/citation/1958/04000/THE_DEVELOPMENT_OF_CONCEPTS_RELATING_TO_THE.1.aspx?casa_token=_No3n1x0l88AAAAA:Q9igp74YnhW437zXlRsNn6N4p1rOQceMG0ZzJ_YR_MVaqxleMWGDFfiK9q7NZxBL_hyRVx8i1JovEdj8muYnDe2EZ0YCg2AF (visited on 11/22/2023).

[28] Romain Brette. "Philosophy of the Spike: Rate-Based vs. Spike-Based Theories of the Brain". In: *Frontiers in Systems Neuroscience* 9 (2015). ISSN: 1662-5137. DOI: 10.3389/fnsys.2015.00151. URL: https://www.frontiersin.org/articles/10.3389/fnsys.2015.00151/full (visited on 05/01/2019).

[29] L. Brodin et al. "Computer Simulations of N-methyl-D-aspartate Receptor-Induced Membrane Properties in a Neuron Model". In: *Journal of Neurophysiology* 66.2 (Aug. 1991), pp. 473–484. ISSN: 0022-3077. DOI: 10.1152/jn.1991.66.2.473. URL: https://journals.physiology.org/doi/abs/10.1152/jn.1991.66.2.473 (visited on 03/14/2022).

[30] H. L. Bryant and J. P. Segundo. "Spike Initiation by Transmembrane Current: A White-Noise Analysis." In: *The Journal of Physiology* 260.2 (1976), pp. 279–314. ISSN: 1469-7793. DOI: 10.1113/jphysiol.1976.sp011516. URL: https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1976.sp011516 (visited on 08/23/2019).

[31] Thomas SJ Burger, Michael E. Rule, and Timothy O'Leary. "Active Dendrites Enable Robust Spiking Computations Despite Timing Jitter". In: *eLife* 12 (Sept. 29, 2023). DOI: 10.7554/eLife.89629. URL: https://elifesciences.org/reviewed-preprints/89629 (visited on 10/26/2023).

[32] Thiago B Burghi. Personal communcations.

[33] Thiago B Burghi and Rodolphe Sepulchre. "Adaptive Observers for Biophysical Neuronal Circuits". In: (2021).

[34] Thiago B. Burghi, Maarten Schoukens, and Rodolphe Sepulchre. "Feedback Identification of Conductance-Based Models". In: *Automatica* 123 (Jan. 1, 2021), p. 109297. ISSN: 0005-1098. DOI: 10.1016/j.automatica.2020.109297. URL: https://www.sciencedirect.com/science/article/pii/S0005109820304969 (visited on 04/29/2021).

[35] Thiago B. Burghi, Maarten Schoukens, and Rodolphe Sepulchre. "System Identification of Biophysical Neuronal Models". In: *2020 59th IEEE Conference on Decision and Control (CDC)*. 2020 59th IEEE Conference on Decision and Control (CDC). Jeju, Korea (South): IEEE, Dec. 14, 2020, pp. 6180–6185. ISBN: 978-1-7281-7447-1. DOI: 10.1109/CDC42340.2020.9304363. URL: https://ieeexplore.ieee.org/document/9304363/ (visited on 09/26/2023).

[36] Daniel A. Butts and Mark S. Goldman. "Tuning Curves, Neuronal Variability, and Sensory Coding". In: *PLOS Biology* 4.4 (Mar. 21, 2006), e92. ISSN: 1545-7885. DOI:

10.1371/journal.pbio.0040092. URL: https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.0040092 (visited on 08/05/2022).

[37] György Buzsáki, Nikos Logothetis, and Wolf Singer. "Scaling Brain Size, Keeping Timing: Evolutionary Preservation of Brain Rhythms". In: *Neuron* 80.3 (Oct. 2013), pp. 751–764. ISSN: 08966273. DOI: 10.1016/j.neuron.2013.10.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S0896627313009045 (visited on 11/22/2023).

[38] György Buzsáki and Xiao-Jing Wang. "Mechanisms of Gamma Oscillations". In: *Annual Review of Neuroscience* 35.1 (July 21, 2012), pp. 203–225. ISSN: 0147-006X, 1545-4126. DOI: 10.1146/annurev-neuro-062111-150444. URL: http://www.annualreviews.org/doi/10.1146/annurev-neuro-062111-150444 (visited on 12/02/2019).

[39] W H Calvin and C F Stevens. "Synaptic Noise and Other Sources of Randomness in Motoneuron Interspike Intervals." In: *Journal of Neurophysiology* 31.4 (July 1968), pp. 574–587. ISSN: 0022-3077. DOI: 10.1152/jn.1968.31.4.574. URL: https://journals.physiology.org/doi/abs/10.1152/jn.1968.31.4.574 (visited on 11/13/2023).

[40] John Certaine. "The Solution of Ordinary Differential Equations with Large Time Constants". In: *Mathematical methods for digital computers* 1 (1960), pp. 128–132.

[41] Mark M. Churchland et al. "Stimulus Onset Quenches Neural Variability: A Widespread Cortical Phenomenon". In: *Nature Neuroscience* 13.3 (Mar. 2010), pp. 369–378. ISSN: 1546-1726. DOI: 10.1038/nn.2501. URL: https://www.nature.com/articles/nn.2501 (visited on 08/24/2019).

[42] Jozsef Csicsvari et al. "Mechanisms of Gamma Oscillations in the Hippocampus of the Behaving Rat". In: *Neuron* 37.2 (Jan. 23, 2003), pp. 311–322. ISSN: 0896-6273. DOI: 10.1016/S0896-6273(02)01169-8. pmid: 12546825. URL: https://www.cell.com/neuron/abstract/S0896-6273(02)01169-8 (visited on 08/24/2023).

[43] Catherine S. Cutts and Stephen J. Eglen. "Detecting Pairwise Correlations in Spike Trains: An Objective Comparison of Methods and Application to the Study of Retinal Waves". In: *Journal of Neuroscience* 34.43 (2014), pp. 14288–14303. URL: https://www.jneurosci.org/content/34/43/14288.short (visited on 09/26/2024).

[44] Simon Davidson and Steve B. Furber. "Comparison of Artificial and Spiking Neural Networks on Digital Hardware". In: *Frontiers in Neuroscience* 15 (2021). ISSN: 1662-453X. DOI: 10.3389/fnins.2021.651141. URL: https://www.frontiersin.org/articles/10.3389/fnins.2021.651141/full (visited on 04/30/2021).

[45] Peter Dayan and L. F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Computational Neuroscience. Cambridge, Mass: Massachusetts Institute of Technology Press, 2001. 460 pp. ISBN: 978-0-262-04199-7.

[46] Alain Destexhe and Denis Paré. “Impact of Network Activity on the Integrative Properties of Neocortical Pyramidal Neurons In Vivo”. In: *Journal of Neurophysiology* 81.4 (Apr. 1, 1999), pp. 1531–1547. ISSN: 0022-3077, 1522-1598. DOI: 10.1152/jn.1999.81.4.1531. URL: https://www.physiology.org/doi/10.1152/jn.1999.81.4.1531 (visited on 11/09/2023).

[47] Alain Destexhe, Michael Rudolph, and Denis Paré. “The High-Conductance State of Neocortical Neurons in Vivo”. In: *Nature Reviews Neuroscience* 4.9 (9 Sept. 2003), pp. 739–751. ISSN: 1471-0048. DOI: 10.1038/nrn1198. URL: https://www.nature.com/articles/nrn1198 (visited on 11/09/2023).

[48] Julie Dethier et al. “A Positive Feedback at the Cellular Level Promotes Robustness and Modulation at the Circuit Level”. In: *Journal of Neurophysiology* 114.4 (Aug. 26, 2015), pp. 2472–2484. ISSN: 0022-3077. DOI: 10.1152/jn.00471.2015. URL: https://www.physiology.org/doi/full/10.1152/jn.00471.2015 (visited on 10/23/2019).

[49] Michael R. DeWeese, Michael Wehr, and Anthony M. Zador. “Binary Spiking in Auditory Cortex”. In: *Journal of Neuroscience* 23.21 (Aug. 27, 2003), pp. 7940–7949. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.23-21-07940.2003. pmid: 12944525. URL: https://www.jneurosci.org/content/23/21/7940 (visited on 08/04/2022).

[50] G. Drion et al. “Neuronal Behaviors: A Control Perspective”. In: *2015 54th IEEE Conference on Decision and Control (CDC)*. 2015 54th IEEE Conference on Decision and Control (CDC). Dec. 2015, pp. 1923–1944. DOI: 10.1109/CDC.2015.7402491.

[51] Guillaume Drion, Alessio Franci, and Rodolphe Sepulchre. “Cellular Switches Orchestrate Rhythmic Circuits”. In: *Biological Cybernetics* 113.1–2 (Apr. 2019), pp. 71–82. ISSN: 0340-1200, 1432-0770. DOI: 10.1007/s00422-018-0778-6. URL: http://link.springer.com/10.1007/s00422-018-0778-6 (visited on 04/08/2019).

[52] G. B. Ermentrout and N. Kopell. “Fine Structure of Neural Spiking and Synchronization in the Presence of Conduction Delays”. In: *Proceedings of the National Academy of Sciences* 95.3 (Feb. 3, 1998), pp. 1259–1264. DOI: 10.1073/pnas.95.3.1259. URL: https://www.pnas.org/doi/abs/10.1073/pnas.95.3.1259 (visited on 08/24/2023).

[53] A. Aldo Faisal, Luc P. J. Selen, and Daniel M. Wolpert. “Noise in the Nervous System”. In: *Nature Reviews Neuroscience* 9.4 (4 Apr. 2008), pp. 292–303. ISSN: 1471-0048. DOI: 10.1038/nrn2258. URL: https://www.nature.com/articles/nrn2258 (visited on 08/04/2022).

[54] Peter Földiák. “The ‘Ideal Homunculus’: Statistical Inference from Neural Population Responses”. In: *Computation and Neural Systems*. Ed. by Frank H. Eeckman and James M. Bower. Boston, MA: Springer US, 1993, pp. 55–60. ISBN: 978-1-4615-3254-5. DOI: 10.1007/978-1-4615-3254-5_9. URL: https://doi.org/10.1007/978-1-4615-3254-5_9 (visited on 08/05/2022).

[55] Alessio Franci, Guillaume Drion, and Rodolphe Sepulchre. "The Sensitivity Function of Excitable Feedback Systems". In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. 2019 IEEE 58th Conference on Decision and Control (CDC). Dec. 2019, pp. 4723–4728. DOI: 10.1109/CDC40024.2019.9029676.

[56] Felix Franke et al. "Structures of Neural Correlation and How They Favor Coding". In: *Neuron* 89.2 (Jan. 20, 2016), pp. 409–422. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2015.12.037. URL: https://www.sciencedirect.com/science/article/pii/S0896627315011393 (visited on 08/04/2022).

[57] Elke C. Fuchs et al. "Recruitment of Parvalbumin-Positive Interneurons Determines Hippocampal Function and Associated Behavior". In: *Neuron* 53.4 (Feb. 15, 2007), pp. 591–604. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2007.01.031. pmid: 17296559. URL: https://www.cell.com/neuron/abstract/S0896-6273(07)00072-4 (visited on 08/24/2023).

[58] Alexander V. Galazyuk and Albert S. Feng. "Encoding of Sound Duration by Neurons in the Auditory Cortex of the Little Brown Bat, Myotis Lucifugus". In: *Journal of Comparative Physiology A* 180.4 (Mar. 1, 1997), pp. 301–311. ISSN: 1432-1351. DOI: 10.1007/s003590050050. URL: https://doi.org/10.1007/s003590050050 (visited on 08/19/2019).

[59] Peng P. Gao et al. "Local Glutamate-Mediated Dendritic Plateau Potentials Change the State of the Cortical Pyramidal Neuron". In: *Journal of Neurophysiology* 125.1 (Jan. 1, 2021), pp. 23–42. ISSN: 0022-3077, 1522-1598. DOI: 10.1152/jn.00734.2019. URL: https://journals.physiology.org/doi/10.1152/jn.00734.2019 (visited on 11/22/2021).

[60] Wulfram Gerstner and Werner M. Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, Aug. 15, 2002. 498 pp. ISBN: 978-0-521-89079-3. Google Books: Rs4oc7HfxIUC.

[61] *GitHub - JuliaDynamics/CausalityTools.jl: Algorithms for detecting associations, dynamical influences and causal inference from data.* GitHub. URL: https://github.com/JuliaDynamics/CausalityTools.jl (visited on 12/19/2023).

[62] Jean-Marc Goaillard et al. "Diversity of Axonal and Dendritic Contributions to Neuronal Output". In: *Frontiers in Cellular Neuroscience* 13 (2020). ISSN: 1662-5102. URL: https://www.frontiersin.org/articles/10.3389/fncel.2019.00570 (visited on 02/14/2023).

[63] Lea Goetz, Arnd Roth, and Michael Häusser. "Active Dendrites Enable Strong but Sparse Inputs to Determine Orientation Selectivity". In: *Proceedings of the National Academy of Sciences* 118.30 (July 27, 2021), e2017339118. DOI: 10.1073/pnas.2017339118. URL: https://www.pnas.org/doi/abs/10.1073/pnas.2017339118 (visited on 07/28/2022).

[64] José Francisco Gómez González, Bartlett Mel, and Panayiota Poirazi. "Distinguishing Linear vs. Non-Linear Integration in CA1 Radial Oblique Dendrites: It's about Time". In: *Frontiers in Computational Neuroscience* 5 (2011). ISSN: 1662-5188. URL: https://www.frontiersin.org/articles/10.3389/fncom.2011.00044 (visited on 08/08/2022).

[65] Robbe L. T. Goris, J. Anthony Movshon, and Eero P. Simoncelli. "Partitioning Neuronal Variability". In: *Nature Neuroscience* 17.6 (6 June 2014), pp. 858–865. ISSN: 1546-1726. DOI: 10.1038/nn.3711. URL: https://www.nature.com/articles/nn.3711 (visited on 01/29/2021).

[66] Donald R. Griffin, Frederic A. Webster, and Charles R. Michael. "The Echolocation of Flying Insects by Bats". In: *Animal Behaviour* 8.3 (July 1, 1960), pp. 141–154. ISSN: 0003-3472. DOI: 10.1016/0003-3472(60)90022-1. URL: http://www.sciencedirect.com/science/article/pii/0003347260900221 (visited on 08/19/2019).

[67] Robert Gütig and Haim Sompolinsky. "The Tempotron: A Neuron That Learns Spike Timing–Based Decisions". In: *Nature Neuroscience* 9.3 (3 Mar. 2006), pp. 420–428. ISSN: 1546-1726. DOI: 10.1038/nn1643. URL: https://www.nature.com/articles/nn1643 (visited on 03/31/2021).

[68] Bilal Haider and David A. McCormick. "Rapid Neocortical Dynamics: Cellular and Network Mechanisms". In: *Neuron* 62.2 (2009), pp. 171–189. URL: https://www.cell.com/neuron/pdf/S0896-6273(09)00289-X.pdf (visited on 11/09/2023).

[69] Norbert Hájos and Ole Paulsen. "Network Mechanisms of Gamma Oscillations in the CA3 Region of the Hippocampus". In: *Neural Networks*. Cortical Microcircuits 22.8 (Oct. 1, 2009), pp. 1113–1119. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2009.07.024. URL: https://www.sciencedirect.com/science/article/pii/S0893608009001713 (visited on 11/09/2023).

[70] H. Helmholtz. *Wissenschaftliche Abhandlungen*. J.A. Barth, 1882. 976 pp. Google Books: veo0AQAAMAAJ.

[71] Mike Hemberger et al. "Reliable Sequential Activation of Neural Assemblies by Single Pyramidal Cells in a Three-Layered Cortex". In: *Neuron* (Aug. 19, 2019). ISSN: 0896-6273. DOI: 10.1016/j.neuron.2019.07.017. URL: http://www.sciencedirect.com/science/article/pii/S0896627319306439 (visited on 10/09/2019).

[72] M Herrmann, J A Hertz, and A Prügel-Bennett. "Analysis of Synfire Chains". In: *Network: Computation in Neural Systems* 6.3 (Jan. 1, 1995), pp. 403–414. ISSN: 0954-898X. DOI: 10.1088/0954-898X_6_3_006. URL: https://doi.org/10.1088/0954-898X_6_3_006 (visited on 03/10/2022).

[73] M. L. Hines and N. T. Carnevale. "The NEURON Simulation Environment". In: *Neural Computation* 9.6 (Aug. 1997), pp. 1179–1209. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.6.1179.

[74] Mark H. Histed and John H. R. Maunsell. “Cortical Neural Populations Can Guide Behavior by Integrating Inputs Linearly, Independent of Synchrony”. In: *Proceedings of the National Academy of Sciences* 111.1 (Jan. 7, 2014), E178–E187. DOI: 10.1073/pnas.1318750111. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1318750111 (visited on 02/14/2023).

[75] A. L. Hodgkin and A. F. Huxley. “A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve”. In: *The Journal of Physiology* 117.4 (Aug. 28, 1952), pp. 500–544. ISSN: 00223751. DOI: 10.1113/jphysiol.1952.sp004764. URL: http://doi.wiley.com/10.1113/jphysiol.1952.sp004764 (visited on 04/30/2019).

[76] A. L. Hodgkin and A. F. Huxley. “Action Potentials Recorded from Inside a Nerve Fibre”. In: *Nature* 144.3651 (3651 Oct. 1939), pp. 710–711. ISSN: 1476-4687. DOI: 10.1038/144710a0. URL: https://www.nature.com/articles/144710a0 (visited on 11/15/2023).

[77] A. L. Hodgkin and A. F. Huxley. “The Dual Effect of Membrane Potential on Sodium Conductance in the Giant Axon of Loligo”. In: *The Journal of Physiology* 116.4 (Apr. 28, 1952), pp. 497–506. ISSN: 0022-3751. pmid: 14946715. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1392212/ (visited on 09/04/2023).

[78] Alan Lloyd Hodgkin, William Albert Hugh Rushton, and Edgar Douglas Adrian. “The Electrical Constants of a Crustacean Nerve Fibre”. In: *Proceedings of the Royal Society of London. Series B - Biological Sciences* 133.873 (Dec. 3, 1946), pp. 444–479. DOI: 10.1098/rspb.1946.0024. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rspb.1946.0024 (visited on 11/17/2023).

[79] D. H. Hubel and T. N. Wiesel. “Receptive Fields of Single Neurones in the Cat’s Striate Cortex”. In: *The Journal of Physiology* 148.3 (1959), pp. 574–591. ISSN: 1469-7793. DOI: 10.1113/jphysiol.1959.sp006308. URL: https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1959.sp006308 (visited on 08/22/2019).

[80] Dongsung Huh and Terrence J. Sejnowski. “Gradient Descent for Spiking Neural Networks”. June 19, 2017. arXiv: 1706.04698 [cs, q-bio, stat]. URL: http://arxiv.org/abs/1706.04698 (visited on 03/31/2021).

[81] Mike Innes et al. *A Differentiable Programming System to Bridge Machine Learning and Scientific Computing*. July 18, 2019. DOI: 10.48550/arXiv.1907.07587. arXiv: 1907.07587 [cs]. URL: http://arxiv.org/abs/1907.07587 (visited on 09/12/2024). Pre-published.

[82] Adam L. Jacobs et al. “Ruling out and Ruling in Neural Codes”. In: *Proceedings of the National Academy of Sciences* 106.14 (Apr. 7, 2009), pp. 5936–5941. DOI: 10.1073/pnas.0900573106. pmid: 19297621. URL: https://www.pnas.org/doi/abs/10.1073/pnas.0900573106 (visited on 02/14/2023).

[83] Sven Jahnke, Raoul-Martin Memmesheimer, and Marc Timme. "Propagating Synchrony in Feed-Forward Networks". In: *Frontiers in Computational Neuroscience* 7 (2013), p. 153. ISSN: 1662-5188. DOI: 10.3389/fncom.2013.00153. URL: https://www.frontiersin.org/article/10.3389/fncom.2013.00153 (visited on 12/06/2021).

[84] Daniel Johnston et al. "Active Properties of Neuronal Dendrites". In: ().

[85] Tim C. Kietzmann, Patrick McClure, and Nikolaus Kriegeskorte. *Deep Neural Networks in Computational Neuroscience*. June 5, 2018. DOI: 10.1101/133504. URL: https://www.biorxiv.org/content/10.1101/133504v2 (visited on 11/15/2023). Pre-published.

[86] H. G. Kim and B. W. Connors. "Apical Dendrites of the Neocortex: Correlation between Sodium- and Calcium-Dependent Spiking and Pyramidal Cell Morphology". In: *Journal of Neuroscience* 13.12 (Dec. 1, 1993), pp. 5301–5311. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.13-12-05301.1993. pmid: 8254376. URL: https://www.jneurosci.org/content/13/12/5301 (visited on 08/05/2022).

[87] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization. Conference Paper at ICLR 2015". 2015. arXiv: 1412.6980.

[88] Werner M. Kistler and Wulfram Gerstner. "Stable Propagation of Activity Pulses in Populations of Spiking Neurons". In: *Neural Computation* 14.5 (May 1, 2002), pp. 987–997. ISSN: 0899-7667. DOI: 10.1162/089976602753633358. URL: https://doi.org/10.1162/089976602753633358 (visited on 03/10/2022).

[89] Christof Koch. *Biophysics of Computation: Information Processing in Single Neurons*. Oxford University Press, Oct. 28, 2004. 588 pp. ISBN: 978-0-19-029285-0. Google Books: aeAJCAAAQBAJ.

[90] Adam Kohn et al. "Correlations and Neuronal Population Information". In: *Annual Review of Neuroscience* 39.1 (July 8, 2016), pp. 237–256. ISSN: 0147-006X, 1545-4126. DOI: 10.1146/annurev-neuro-070815-013851. pmid: 27145916. URL: http://www.annualreviews.org/doi/10.1146/annurev-neuro-070815-013851 (visited on 08/24/2019).

[91] Matthew E. Larkum, J. Julius Zhu, and Bert Sakmann. "A New Cellular Mechanism for Coupling Inputs Arriving at Different Cortical Layers". In: *Nature* 398.6725 (Mar. 1999), pp. 338–341. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/18686. URL: http://www.nature.com/articles/18686 (visited on 07/11/2021).

[92] Jean-François Léger et al. "Synaptic Integration in Rat Frontal Cortex Shaped by Network Activity". In: *Journal of Neurophysiology* 93.1 (Jan. 2005), pp. 281–293. ISSN: 0022-3077, 1522-1598. DOI: 10.1152/jn.00067.2003. URL: https://www.physiology.org/doi/10.1152/jn.00067.2003 (visited on 11/10/2023).

[93] J. Y. Lettvin et al. “What the Frog’s Eye Tells the Frog’s Brain”. In: *Proceedings of the IRE* 47.11 (Nov. 1959), pp. 1940–1951. ISSN: 2162-6634. DOI: 10.1109/JRPROC.1959.287207. URL: https://ieeexplore.ieee.org/abstract/document/4065609 (visited on 11/17/2023).

[94] L. S. Leung. “Nonlinear Feedback Model of Neuronal Populations in Hippocampal CAl Region”. In: *Journal of Neurophysiology* 47.5 (May 1982), pp. 845–868. ISSN: 0022-3077. DOI: 10.1152/jn.1982.47.5.845. URL: https://journals.physiology.org/doi/abs/10.1152/jn.1982.47.5.845 (visited on 08/24/2023).

[95] William B Levy and Robert A. Baxter. “Energy Efficient Neural Codes”. In: *Neural Computation* 8.3 (Apr. 1, 1996), pp. 531–543. ISSN: 0899-7667. DOI: 10.1162/neco.1996.8.3.531. URL: https://doi.org/10.1162/neco.1996.8.3.531 (visited on 11/20/2023).

[96] Yonatan Loewenstein and Haim Sompolinsky. “Temporal Integration by Calcium Dynamics in a Model Neuron”. In: *Nature Neuroscience* 6.9 (9 Sept. 2003), pp. 961–967. ISSN: 1546-1726. DOI: 10.1038/nn1109. URL: https://www.nature.com/articles/nn1109 (visited on 02/14/2023).

[97] Michael London and Michael Häusser. “Dendritic Computation”. In: *Annual Review of Neuroscience* 28.1 (2005), pp. 503–532. DOI: 10.1146/annurev.neuro.28.061604.135703. pmid: 16033324. URL: https://doi.org/10.1146/annurev.neuro.28.061604.135703 (visited on 08/05/2022).

[98] Michael London et al. “Sensitivity to Perturbations in Vivo Implies High Noise and Suggests Rate Coding in Cortex”. In: *Nature* 466.7302 (7302 July 2010), pp. 123–127. ISSN: 1476-4687. DOI: 10.1038/nature09086. URL: https://www.nature.com/articles/nature09086 (visited on 05/06/2022).

[99] Attila Losonczy and Jeffrey C. Magee. “Integrative Properties of Radial Oblique Dendrites in Hippocampal CA1 Pyramidal Neurons”. In: *Neuron* 50.2 (Apr. 20, 2006), pp. 291–307. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2006.03.016. URL: https://www.sciencedirect.com/science/article/pii/S0896627306002133 (visited on 02/14/2023).

[100] *LsqFit.Jl*. JuliaNLSolvers, Aug. 30, 2023. URL: https://github.com/JuliaNLSolvers/LsqFit.jl (visited on 09/11/2023).

[101] Oleg Borisovich Lupanov. “Circuits using threshold elements”. In: *Doklady Akademii Nauk* 202.6 (1972), pp. 1288–1291.

[102] W. Maass, G. Schnitger, and E.D. Sontag. “On the Computational Power of Sigmoid versus Boolean Threshold Circuits”. In: *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*. 32nd Annual Symposium of Foundations of Computer Science. San Juan, Puerto Rico: IEEE Comput. Soc. Press, 1991, pp. 767–

776. ISBN: 978-0-8186-2445-2. DOI: 10.1109/SFCS.1991.185447. URL: http://ieeexplore.ieee.org/document/185447/ (visited on 02/07/2023).

[103] Wolfgang Maass. "Networks of Spiking Neurons: The Third Generation of Neural Network Models". In: *Neural Networks* 10.9 (Dec. 1, 1997), pp. 1659–1671. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(97)00011-7. URL: http://www.sciencedirect.com/science/article/pii/S0893608097000117 (visited on 03/14/2019).

[104] Z. Mainen and T. Sejnowski. "Reliability of Spike Timing in Neocortical Neurons". In: *Science* 268.5216 (June 9, 1995), pp. 1503–1506. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.7770778. URL: http://www.sciencemag.org/cgi/doi/10.1126/science.7770778 (visited on 06/10/2019).

[105] Guy Major, Matthew E. Larkum, and Jackie Schiller. "Active Properties of Neocortical Pyramidal Neuron Dendrites". In: *Annual Review of Neuroscience* 36.1 (July 8, 2013), pp. 1–24. ISSN: 0147-006X, 1545-4126. DOI: 10.1146/annurev-neuro-062111-150343. URL: https://www.annualreviews.org/doi/10.1146/annurev-neuro-062111-150343 (visited on 06/24/2022).

[106] Guy Major et al. "Spatiotemporally Graded NMDA Spike/Plateau Potentials in Basal Dendrites of Neocortical Pyramidal Neurons". In: *Journal of Neurophysiology* 99.5 (May 2008), pp. 2584–2601. ISSN: 0022-3077, 1522-1598. DOI: 10.1152/jn.00011.2008. URL: https://www.physiology.org/doi/10.1152/jn.00011.2008 (visited on 07/03/2021).

[107] Judit K. Makara and Jeffrey C. Magee. "Variable Dendritic Integration in Hippocampal CA3 Pyramidal Neurons". In: *Neuron* 80.6 (Dec. 18, 2013), pp. 1438–1450. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2013.10.033. URL: https://www.sciencedirect.com/science/article/pii/S0896627313009859 (visited on 02/14/2023).

[108] Adam H. Marblestone, Greg Wayne, and Konrad P. Kording. "Toward an Integration of Deep Learning and Neuroscience". In: *Frontiers in Computational Neuroscience* 10 (2016). ISSN: 1662-5188. URL: https://www.frontiersin.org/articles/10.3389/fncom.2016.00094 (visited on 11/15/2023).

[109] Eve Marder. "Variability, Compensation, and Modulation in Neurons and Circuits". In: *Proceedings of the National Academy of Sciences* 108 (supplement_3 Sept. 13, 2011), pp. 15542–15548. DOI: 10.1073/pnas.1010674108. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1010674108 (visited on 08/08/2023).

[110] Eve Marder and Jean-Marc Goaillard. "Variability, Compensation and Homeostasis in Neuron and Network Function". In: *Nature Reviews Neuroscience* 7.7 (7 July 2006), pp. 563–574. ISSN: 1471-0048. DOI: 10.1038/nrn1949. URL: https://www.nature.com/articles/nrn1949 (visited on 08/08/2023).

[111] Mackenzie Weygandt Mathis and Alexander Mathis. "Deep Learning Tools for the Measurement of Animal Behavior in Neuroscience". In: *Current Opinion in*

Neurobiology. Neurobiology of Behavior 60 (Feb. 1, 2020), pp. 1–11. ISSN: 0959-4388. DOI: 10.1016/j.conb.2019.10.008. URL: https://www.sciencedirect.com/science/article/pii/S0959438819301151 (visited on 11/15/2023).

[112] Warren S. McCulloch and Walter Pitts. "A Logical Calculus of the Ideas Immanent in Nervous Activity". In: *The bulletin of mathematical biophysics* 5.4 (Dec. 1, 1943), pp. 115–133. ISSN: 1522-9602. DOI: 10.1007/BF02478259. URL: https://doi.org/10.1007/BF02478259 (visited on 11/14/2023).

[113] C. Mead. "Neuromorphic Electronic Systems". In: *Proceedings of the IEEE* 78.10 (Oct. 1990), pp. 1629–1636. ISSN: 1558-2256. DOI: 10.1109/5.58356. URL: https://ieeexplore.ieee.org/abstract/document/58356?casa_token=BHhz9AO0bi4AAAAA:1yGkv7WdM3ruNJPrARlrVC6ytTzrRS8i25p7EeCbLQu8xiodaTVWjFMnE8HzESiqOIOuxqdM (visited on 11/22/2023).

[114] Julien Offray de La Mettrie. *L'homme machine*. Luzac, 1748. 138 pp. Google Books: JJ1b00HkWp4C.

[115] Cristopher Moore and Stephan Mertens. *The Nature of Computation*. OUP Oxford, Aug. 11, 2011. 631 pp. ISBN: 978-0-19-162080-5. Google Books: jnGKbpMV8xoC.

[116] Peter M. Narins and Edwin R. Lewis. "The Vertebrate Ear as an Exquisite Seismic Sensor". In: *The Journal of the Acoustical Society of America* 76.5 (Nov. 1, 1984), pp. 1384–1387. ISSN: 0001-4966. DOI: 10.1121/1.391455. URL: https://doi.org/10.1121/1.391455 (visited on 11/19/2023).

[117] Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. "Surrogate Gradient Learning in Spiking Neural Networks". May 3, 2019. arXiv: 1901.09948 [cs, q-bio]. URL: http://arxiv.org/abs/1901.09948 (visited on 09/17/2020).

[118] John O'Keefe and Michael L. Recce. "Phase Relationship between Hippocampal Place Units and the EEG Theta Rhythm". In: *Hippocampus* 3.3 (1993), pp. 317–330. ISSN: 1098-1063. DOI: 10.1002/hipo.450030307. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/hipo.450030307 (visited on 08/19/2019).

[119] Timothy O'Leary and David J.A. Wyllie. "Single-Channel Properties of N-methyl-D-aspartate Receptors Containing Chimaeric GluN2A/GluN2D Subunits". In: *Biochemical Society Transactions* 37.6 (Nov. 19, 2009), pp. 1347–1354. ISSN: 0300-5127. DOI: 10.1042/BST0371347. URL: https://doi.org/10.1042/BST0371347 (visited on 02/14/2023).

[120] J. Park and I.W. Sandberg. "Criteria for the Approximation of Nonlinear Systems". In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 39.8 (Aug. 1992), pp. 673–676. ISSN: 10577122. DOI: 10.1109/81.168920. URL: http://ieeexplore.ieee.org/document/168920/ (visited on 09/26/2023).

[121] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the Difficulty of Training Recurrent Neural Networks". In: *Proceedings of the 30th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, May 26, 2013, pp. 1310–1318. URL: https://proceedings.mlr.press/v28/pascanu13.html (visited on 12/23/2023).

[122] Donald H. Perkel, George L. Gerstein, and George P. Moore. "Neuronal Spike Trains and Stochastic Point Processes: I. The Single Spike Train". In: *Biophysical Journal* 7.4 (July 1, 1967), pp. 391–418. ISSN: 0006-3495. DOI: 10.1016/S0006-3495(67)86596-2. URL: http://www.sciencedirect.com/science/article/pii/S0006349567865962 (visited on 08/20/2019).

[123] D. I. Perrett, E. T. Rolls, and W. Caan. "Visual Neurones Responsive to Faces in the Monkey Temporal Cortex". In: *Experimental Brain Research* 47.3 (Sept. 1, 1982), pp. 329–342. ISSN: 1432-1106. DOI: 10.1007/BF00239352. URL: https://doi.org/10.1007/BF00239352 (visited on 08/19/2019).

[124] Panayiota Poirazi, Terrence Brannon, and Bartlett W. Mel. "Pyramidal Neuron as Two-Layer Neural Network". In: *Neuron* 37.6 (Mar. 27, 2003), pp. 989–999. ISSN: 0896-6273. DOI: 10.1016/S0896-6273(03)00149-1. URL: https://www.sciencedirect.com/science/article/pii/S0896627303001491 (visited on 02/14/2023).

[125] Alon Poleg-Polsky. "Dendritic Spikes Expand the Range of Well Tolerated Population Noise Structures". In: *The Journal of Neuroscience* 39.46 (Nov. 13, 2019), pp. 9173–9184. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.0638-19.2019. URL: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.0638-19.2019 (visited on 08/04/2022).

[126] Alon Polsky, Bartlett W. Mel, and Jackie Schiller. "Computational Subunits in Thin Dendrites of Pyramidal Cells". In: *Nature Neuroscience* 7.6 (6 June 2004), pp. 621–627. ISSN: 1546-1726. DOI: 10.1038/nn1253. URL: https://www.nature.com/articles/nn1253 (visited on 02/14/2023).

[127] Wilfrid Rall. "Branching Dendritic Trees and Motoneuron Membrane Resistivity". In: *Experimental Neurology* 1.5 (Nov. 1, 1959), pp. 491–527. ISSN: 0014-4886. DOI: 10.1016/0014-4886(59)90046-9. URL: https://www.sciencedirect.com/science/article/pii/0014488659900469 (visited on 11/17/2023).

[128] Luka Ribar and Rodolphe Sepulchre. "Neuromodulation of Neuromorphic Circuits". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 66.8 (Aug. 2019), pp. 3028–3040. ISSN: 1558-0806. DOI: 10.1109/TCSI.2019.2907113.

[129] Blake A. Richards et al. "A Deep Learning Framework for Neuroscience". In: *Nature Neuroscience* 22.11 (11 Nov. 2019), pp. 1761–1770. ISSN: 1546-1726. DOI: 10.1038/s41593-019-0520-2. URL: https://www.nature.com/articles/s41593-019-0520-2 (visited on 11/15/2023).

[130] Fred Rieke et al. *Spikes: Exploring the Neural Code*. Cambridge, MA, USA: MIT Press, 1999. ISBN: 978-0-262-18174-7.

[131] Alan Roberts and Brian M. H. Bush. *Neurones Without Impulses: Their Significance for Vertebrate and Invertebrate Nervous Systems*. Cambridge University Press, Feb. 5, 1981. 308 pp. ISBN: 978-0-521-29935-0. Google Books: UsuNu9a8XSgC.

[132] Edmund T. Rolls and Tovee, Martin J. "Processing Speed in the Cerebral Cortex and the Neurophysiology of Visual Masking". In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 257.1348 (July 22, 1994), pp. 9–15. DOI: 10.1098/rspb.1994.0087. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rspb.1994.0087 (visited on 08/19/2019).

[133] J E Rose et al. "Phase-Locked Response to Low-Frequency Tones in Single Auditory Nerve Fibers of the Squirrel Monkey." In: *Journal of Neurophysiology* 30.4 (July 1967), pp. 769–793. ISSN: 0022-3077. DOI: 10.1152/jn.1967.30.4.769. URL: https://journals.physiology.org/doi/abs/10.1152/jn.1967.30.4.769 (visited on 11/19/2023).

[134] I.W. Sandberg. "Approximation Theorems for Discrete-Time Systems". In: *[1991] Proceedings of the 34th Midwest Symposium on Circuits and Systems*. 34th Midwest Symposium on Circuits and Systems. Monterey, CA, USA: IEEE, 1992, pp. 6–7. ISBN: 978-0-7803-0620-2. DOI: 10.1109/MWSCAS.1991.252115. URL: http://ieeexplore.ieee.org/document/252115/ (visited on 09/26/2023).

[135] T. D. Sanger. "Probability Density Estimation for the Interpretation of Neural Population Codes". In: *Journal of Neurophysiology* 76.4 (Oct. 1996), pp. 2790–2793. ISSN: 0022-3077. DOI: 10.1152/jn.1996.76.4.2790. URL: https://journals.physiology.org/doi/abs/10.1152/jn.1996.76.4.2790 (visited on 08/05/2022).

[136] Jackie Schiller and Yitzhak Schiller. "NMDA Receptor-Mediated Dendritic Spikes and Coincident Signal Amplification". In: *Current Opinion in Neurobiology* 11.3 (June 1, 2001), pp. 343–348. ISSN: 0959-4388. DOI: 10.1016/S0959-4388(00)00217-8. URL: https://www.sciencedirect.com/science/article/pii/S0959438800002178 (visited on 03/10/2022).

[137] Jackie Schiller et al. "Calcium Action Potentials Restricted to Distal Apical Dendrites of Rat Neocortical Pyramidal Neurons". In: *The Journal of Physiology* 505.3 (1997), pp. 605–616. ISSN: 1469-7793. DOI: 10.1111/j.1469-7793.1997.605ba.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-7793.1997.605ba.x (visited on 08/05/2022).

[138] Jackie Schiller et al. "NMDA Spikes in Basal Dendrites of Cortical Pyramidal Neurons". In: *Nature* 404.6775 (6775 Mar. 2000), pp. 285–289. ISSN: 1476-4687. DOI: 10.1038/35005094. URL: https://www.nature.com/articles/35005094 (visited on 03/08/2022).

[139] Christoph Schmidt-Hieber et al. "Active Dendritic Integration as a Mechanism for Robust and Precise Grid Cell Firing". In: *Nature Neuroscience* 20.8 (8 Aug. 2017), pp. 1114–1121. ISSN: 1546-1726. DOI: 10.1038/nn.4582. URL: https://www.nature.com/articles/nn.4582 (visited on 08/05/2022).

[140] Biswa Sengupta et al. "Action Potential Energy Efficiency Varies Among Neuron Types in Vertebrates and Invertebrates". In: *PLOS Computational Biology* 6.7 (July 1, 2010), e1000840. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1000840. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000840 (visited on 11/20/2023).

[141] R. Sepulchre, G. Drion, and A. Franci. "Control Across Scales by Positive and Negative Feedback". In: *Annual Review of Control, Robotics, and Autonomous Systems* 2.1 (2019), pp. 89–113. DOI: 10.1146/annurev-control-053018-023708. URL: https://doi.org/10.1146/annurev-control-053018-023708 (visited on 11/07/2019).

[142] H S Seung and H Sompolinsky. "Simple Models for Reading Neuronal Population Codes." In: *Proceedings of the National Academy of Sciences* 90.22 (Nov. 15, 1993), pp. 10749–10753. DOI: 10.1073/pnas.90.22.10749. URL: https://www.pnas.org/doi/abs/10.1073/pnas.90.22.10749 (visited on 08/05/2022).

[143] Michael N. Shadlen and William T. Newsome. "The Variable Discharge of Cortical Neurons: Implications for Connectivity, Computation, and Information Coding". In: *Journal of Neuroscience* 18.10 (May 15, 1998), pp. 3870–3896. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.18-10-03870.1998. pmid: 9570816. URL: https://www.jneurosci.org/content/18/10/3870 (visited on 08/20/2019).

[144] C. E. Shannon. "A Mathematical Theory of Communication". In: *The Bell System Technical Journal* 27.3 (July 1948), pp. 379–423. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1948.tb01338.x. URL: https://ieeexplore.ieee.org/abstract/document/6773024 (visited on 11/20/2023).

[145] Shy Shoham, Daniel H. O'Connor, and Ronen Segev. "How Silent Is the Brain: Is There a "Dark Matter" Problem in Neuroscience?" In: *Journal of Comparative Physiology A* 192.8 (Aug. 1, 2006), pp. 777–784. ISSN: 1432-1351. DOI: 10.1007/s00359-006-0117-6. URL: https://doi.org/10.1007/s00359-006-0117-6 (visited on 11/20/2023).

[146] Jiří Šíma and Pekka Orponen. "General-Purpose Computation with Neural Networks: A Survey of Complexity Theoretic Results". In: *Neural Computation* 15.12 (Dec. 1, 2003), pp. 2727–2778. ISSN: 0899-7667, 1530-888X. DOI: 10.1162/089976603322518731. URL: https://direct.mit.edu/neco/article/15/12/2727-2778/6791 (visited on 02/07/2023).

[147] W. R. Softky and C. Koch. "The Highly Irregular Firing of Cortical Cells Is Inconsistent with Temporal Integration of Random EPSPs". In: *Journal of Neuroscience*

13.1 (Jan. 1, 1993), pp. 334–350. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.13-01-00334.1993. pmid: 8423479. URL: https://www.jneurosci.org/content/13/1/334 (visited on 08/20/2019).

[148] W. A. Spencer and E. R. Kandel. "Electrophysiology of Hippocampal Neurons: Iv. Fast Prepotentials". In: *Journal of Neurophysiology* 24.3 (May 1961), pp. 272–285. ISSN: 0022-3077. DOI: 10.1152/jn.1961.24.3.272. URL: https://journals.physiology.org/doi/abs/10.1152/jn.1961.24.3.272 (visited on 08/05/2022).

[149] *Spytorch/Notebooks/SpyTorchTutorial1.Ipynb at Main · Fzenke/Spytorch*. GitHub. URL: https://github.com/fzenke/spytorch/blob/main/notebooks/SpyTorchTutorial1.ipynb (visited on 10/29/2023).

[150] Peter Sterling and Simon Laughlin. *Principles of Neural Design*. Cambridge, Massachusetts: The MIT Press, 2015. 542 pp. ISBN: 978-0-262-02870-7.

[151] B L Strehler and R Lestienne. "Evidence on Precise Time-Coded Symbols and Memory of Patterns in Monkey Cortical Neuronal Spike Trains." In: *Proceedings of the National Academy of Sciences* 83.24 (Dec. 1986), pp. 9812–9816. DOI: 10.1073/pnas.83.24.9812. URL: https://www.pnas.org/doi/abs/10.1073/pnas.83.24.9812 (visited on 11/20/2023).

[152] Naoya Takahashi et al. "Locally Synchronized Synaptic Inputs". In: *Science* 335.6066 (Jan. 20, 2012), pp. 353–356. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1210362. URL: https://www.science.org/doi/10.1126/science.1210362 (visited on 06/20/2022).

[153] M.C. Teich. "Fractal Character of the Auditory Neural Spike Train". In: *IEEE Transactions on Biomedical Engineering* 36.1 (Jan. 1989), pp. 150–160. ISSN: 1558-2531. DOI: 10.1109/10.16460.

[154] Simon J. Thorpe and Michel Imbert. "Biological Constraints on Connectionist Modelling". In: *Connectionism in Perspective*. Elsevier, 1989, pp. 63–92.

[155] Paul Tiesinga and Terrence J. Sejnowski. "Cortical Enlightenment: Are Attentional Gamma Oscillations Driven by ING or PING?" In: *Neuron* 63.6 (Sept. 2009), pp. 727–732. ISSN: 08966273. DOI: 10.1016/j.neuron.2009.09.009. URL: https://linkinghub.elsevier.com/retrieve/pii/S0896627309006916 (visited on 08/24/2023).

[156] Paul H.E. Tiesinga et al. "Computational Model of Carbachol-Induced Delta, Theta, and Gamma Oscillations in the Hippocampus". In: *Hippocampus* 11.3 (2001), pp. 251–274. ISSN: 1098-1063. DOI: 10.1002/hipo.1041. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/hipo.1041 (visited on 08/29/2023).

[157] D. J. Tolhurst, J. A. Movshon, and A. F. Dean. "The Statistical Reliability of Signals in Single Neurons in Cat and Monkey Visual Cortex". In: *Vision Research* 23.8 (Jan. 1, 1983), pp. 775–785. ISSN: 0042-6989. DOI: 10.1016/0042-6989(83)90200-6.

URL: http://www.sciencedirect.com/science/article/pii/0042698983902006 (visited on 08/20/2019).

[158] Tomas Van Pottelbergh, Guillaume Drion, and Rodolphe Sepulchre. "From Biophysical to Integrate-and-Fire Modeling". In: *Neural Computation* 33.3 (2021), pp. 563–589.

[159] Tomas Van Pottelbergh, Guillaume Drion, and Rodolphe Sepulchre. "Robust Modulation of Integrate-and-Fire Models". In: *Neural Computation* 30.4 (Jan. 30, 2018), pp. 987–1011. ISSN: 0899-7667. DOI: 10.1162/neco_a_01065. URL: https://doi.org/10.1162/neco_a_01065 (visited on 07/11/2019).

[160] Rufin VanRullen and Simon J Thorpe. "Is It a Bird? Is It a Plane? Ultra-Rapid Visual Categorisation of Natural and Artifactual Objects". In: *Perception* 30.6 (June 1, 2001), pp. 655–668. ISSN: 0301-0066. DOI: 10.1068/p3029. URL: https://doi.org/10.1068/p3029 (visited on 03/10/2022).

[161] Aram Vartanian and P. Weiner. "Dictionary of the History of Ideas". In: *Charles Scribners Sons, NY* 4 (1973), p. 307.

[162] Freeman J. Walter. *Mass Action in the Nervous System: Examination of the Neurophysiological Basis of Adaptive Behavior Through the EEG*. Academic Press, 1975. 520 pp. ISBN: 978-0-12-267150-0.

[163] Xiao-Jing Wang. "Neurophysiological and Computational Principles of Cortical Rhythms in Cognition". In: *Physiological Reviews* 90.3 (July 2010), pp. 1195–1268. ISSN: 1522-1210. DOI: 10.1152/physrev.00035.2008. pmid: 20664082.

[164] Dong-Sheng Wei et al. "Compartmentalized and Binary Behavior of Terminal Dendrites in Hippocampal Pyramidal Neurons". In: *Science* 293.5538 (Sept. 21, 2001), pp. 2272–2275. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1061198. pmid: 11567143. URL: https://science.sciencemag.org/content/293/5538/2272 (visited on 07/12/2021).

[165] Gerhard Werner and Vernon B. Mountcastle. "THE VARIABILITY OF CENTRAL NEURAL ACTIVITY IN A SENSORY SYSTEM, AND ITS IMPLICATIONS FOR THE CENTRAL REFLECTION OF SENSORY EVENTS". In: *Journal of Neurophysiology* 26.6 (Nov. 1, 1963), pp. 958–977. ISSN: 0022-3077, 1522-1598. DOI: 10.1152/jn.1963.26.6.958. URL: https://www.physiology.org/doi/10.1152/jn.1963.26.6.958 (visited on 11/18/2023).

[166] M. A Whittington et al. "Inhibition-Based Rhythms: Experimental and Mathematical Observations on Network Dynamics". In: *International Journal of Psychophysiology* 38.3 (Dec. 1, 2000), pp. 315–336. ISSN: 0167-8760. DOI: 10.1016/S0167-8760(00)00173-2. URL: http://www.sciencedirect.com/science/article/pii/S0167876000001732 (visited on 09/24/2019).

[167] Miles A. Whittington, Roger D. Traub, and Natalie E. Adams. "A Future for Neuronal Oscillation Research". In: *Brain and Neuroscience Advances* 2 (Jan. 1, 2018), p. 2398212818794827. ISSN: 2398-2128. DOI: 10.1177/2398212818794827. URL: https://doi.org/10.1177/2398212818794827 (visited on 09/06/2022).

[168] Norbert Wiener. "Cybernetics". In: *Scientific American* 179.5 (1948), pp. 14–19. ISSN: 0036-8733. JSTOR: 24945913. URL: https://www.jstor.org/stable/24945913 (visited on 11/15/2023).

[169] Ronald J. Williams and David Zipser. "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks". In: *Neural Computation* 1.2 (June 1, 1989), pp. 270–280. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.2.270. URL: https://doi.org/10.1162/neco.1989.1.2.270 (visited on 11/01/2023).

[170] H. R. Wilson and J. D. Cowan. "A Mathematical Theory of the Functional Dynamics of Cortical and Thalamic Nervous Tissue". In: *Kybernetik* 13.2 (Sept. 1, 1973), pp. 55–80. ISSN: 1432-0770. DOI: 10.1007/BF00288786. URL: https://doi.org/10.1007/BF00288786 (visited on 08/15/2024).

[171] Hugh R. Wilson and Jack D. Cowan. "Excitatory and Inhibitory Interactions in Localized Populations of Model Neurons". In: *Biophysical Journal* 12.1 (Jan. 1, 1972), pp. 1–24. ISSN: 0006-3495. DOI: 10.1016/S0006-3495(72)86068-5. URL: http://www.sciencedirect.com/science/article/pii/S0006349572860685 (visited on 10/01/2019).

[172] Friedemann Zenke and Surya Ganguli. "SuperSpike: Supervised Learning in Multilayer Spiking Neural Networks". In: *Neural Computation* 30.6 (June 1, 2018), pp. 1514–1541. ISSN: 0899-7667. DOI: 10.1162/neco_a_01086. URL: https://doi.org/10.1162/neco_a_01086 (visited on 06/17/2022).

[173] Friedemann Zenke and Tim P. Vogels. "The Remarkable Robustness of Surrogate Gradient Learning for Instilling Complex Function in Spiking Neural Networks". In: *bioRxiv* (June 29, 2020), p. 2020.06.29.176925. DOI: 10.1101/2020.06.29.176925. URL: https://www.biorxiv.org/content/10.1101/2020.06.29.176925v1 (visited on 09/29/2020).

[174] Jennifer L. Zick et al. "Disparate Insults Relevant to Schizophrenia Converge on Impaired Spike Synchrony and Weaker Synaptic Interactions in Prefrontal Local Circuits". In: *Current Biology* 32.1 (Jan. 2022), 14–25.e4. ISSN: 09609822. DOI: 10.1016/j.cub.2021.10.009. URL: https://linkinghub.elsevier.com/retrieve/pii/S0960982221013634 (visited on 05/10/2022).

[175] E. Zohary, P. Hillman, and S. Hochstein. "Time Course of Perceptual Discrimination and Single Neuron Reliability". In: *Biological Cybernetics* 62.6 (Apr. 1, 1990), pp. 475–486. ISSN: 1432-0770. DOI: 10.1007/BF00205109. URL: https://doi.org/10.1007/BF00205109 (visited on 11/19/2023).

[176] Joel Zylberberg et al. “Robust Information Propagation through Noisy Neural Circuits”. In: *PLOS Computational Biology* 13.4 (Apr. 18, 2017), e1005497. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1005497. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005497 (visited on 08/04/2022).