



UNIVERSITY OF
CAMBRIDGE

Department of Engineering

A Data-Driven Model Of Mouse Visual Cortex

Author Name: Chulabutra Chuenchoksan

Supervisor: Dr Yashar Ahmadian Tehrani

Date: 27th May 2024

I hereby declare that, except where specifically indicated, the work submitted herein is my own original work.

Signed C. Chuenchoksan date 27/05/2024

A Data-Driven Model Of Mouse Visual Cortex

Chulabutra Chuenchoksan, Peterhouse

Technical Abstract

The primary visual cortex (V1) is the region of the brain located in the occipital lobe that is responsible for processing visual information.

It is currently not evident to what extent recurrent connectivity rules explain the orientation selectivity as well as the heterogeneity of neurons in the V1. Two models of the V1 were previously proposed and thoroughly explored in this project. One with random untuned connectivity and one with a non-random orientation-tuned connectivity. Even though both models are able to reproduce the orientation selectivity and contrast invariance property of the neuron, the two studies show conflict in connectivity rules, balance of excitation and inhibition as well as methodology in the investigation of model properties.

This study addresses the conflicts in V1 research and proposes a new data-driven model with a random orientation-tuned recurrent connectivity which could operate in both tight and loose balance of excitation and inhibition.

Different methodologies including Exponential Natural Evolution Strategies (xNES) and gradient descent with backward and forward differentiation were explored to develop a strong framework to fit V1 models to data. The experiment indicates that for smaller neuron populations, gradient descent with backward differentiation is effective, while xNES is more useful for larger neuron populations due to its lower memory usage.

By fitting the model to both simulated and actual data, numerous local minima were observed. Additionally, a bi-modal behaviour of the parameter density and balance of excitation and inhibition were identified that produced statistically similar tuning curves to the data. Because V1 in rodents such as mice lacks a functional map, the results found give rise to the hypothesis that in a mouse, V1 operate with a tighter balance and untuned connectivity, potentially explaining the high heterogeneity across mouse V1 neurons. On the other hand, in animals with a functional map, V1 operates with a looser balance and a more orientation-tuned connectivity.

Lastly, a Convolutional Neural Network-based model was proposed to estimate parameters directly from tuning curves. Initial results show a promising trend indicating the potential of deep learning approaches in this domain.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Yashar Ahmadian Tehrani, for his unwavering support and invaluable guidance throughout this project. He has guided me through the complexities of Computational Neuroscience, a field entirely new to me at the start of this incredible journey. Most importantly, He has provided the most insightful feedback during our weekly meetings which shapes the project in an interesting direction.

I also wish to extend my heartfelt appreciation to my parents for their care and support at every stage of my life. Furthermore, I am immensely grateful to my friends for the memorable experiences we have shared at university.

Contents

1	Introduction	1
1.1	The Primary Visual Cortex	1
1.2	Project Motivation and Objectives	2
2	Background	3
2.1	Balance of Excitation and Inhibition	3
2.2	Previous Studies of V1	4
2.2.1	Random Untuned Connectivity	4
2.2.2	Non-Random Orientation-Tuned Connectivity	6
3	Methods	9
3.1	Proposed V1 Model	9
3.1.1	Mean-Field Equations and The Ricciardi Transfer Function	9
3.1.2	Recurrent Connectivity	10
3.1.3	Feed-Forward Input	12
3.1.4	Fixed-Point Solver	12
3.2	Loss Function and Parameter Constraints	13
3.2.1	Maximum Mean Discrepancy	13
3.2.2	Tuning Curve Normalisation	13
3.2.3	Assumptions and Parameter Constraints	14
3.2.4	Overall Loss Function	15
3.3	Optimisation	16
3.3.1	Gradient Descent	16
3.3.2	Exponential Natural Evolution Strategy	16
3.4	Experimental Datasets	17
3.5	CNN Based Approach	17
4	Results	19
4.1	Method Validation	19
4.1.1	Gradient Descent with Small Neuronal Population	19
4.1.2	Varying sample size	22
4.1.3	High Neuron Population	23
4.2	Fitting Model to Actual V1 Responses	25
4.3	CNN Preliminary Results	28
5	Conclusion and Future Directions	30
5.1	Conclusion	30
5.2	Future Directions	31
Appendix A	Supporting Results	35
A.1	Results From a Non-Random Orientation-Tuned Connectivity	35
A.2	Tuning Curves Examples	36
A.3	Neuronal Activity	39
Appendix B	Risk Assessment	41

Chapter 1

Introduction

1.1 The Primary Visual Cortex

The primary visual cortex (V1) is the first stage of processing in the visual cortex. Its main function is to process visual stimulus signals which flow from the retina through the Lateral Geniculate Nucleus (LGN) to the V1 [1]. This signal path is shown in Figure 1.1.

V1 neurons have been shown to possess orientation selectivity [2]. Orientation selectivity refers to the neuronal property where neurons respond more strongly when presented with a grating stimulus at a specific orientation called the preferred orientation. A grating stimulus is an alternating black-and-white pattern image where the contrast and orientation can vary as shown in Figure 1.1.

The plot of the neuronal response at various grating orientations is called a tuning curve. Figure 1.2 demonstrates a typical V1 tuning curve with orientation selectivity.

Furthermore, the V1 is known to be connected recurrently. This means that the neurons in a certain layer are also connected to neurons within the same layer [1]. The recurrent connection plays a crucial role in governing the properties of the V1 [3].

In animals such as cats and monkeys, the V1 neurons are arranged in a columnar structure which refers to the neurons with similar preferred orientations being physically close together in the network. However, in rodents such as rats and mice, the V1 neurons are arranged without a functional map ie. in a salt-and-pepper manner with neurons being connected randomly independent of their preferred orientation [4].

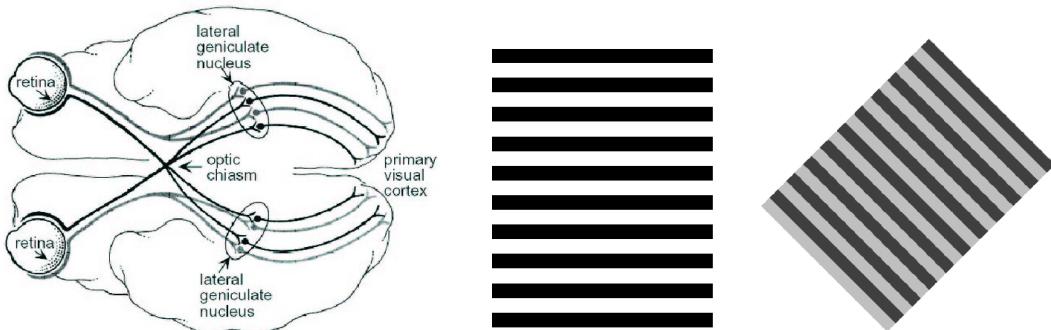
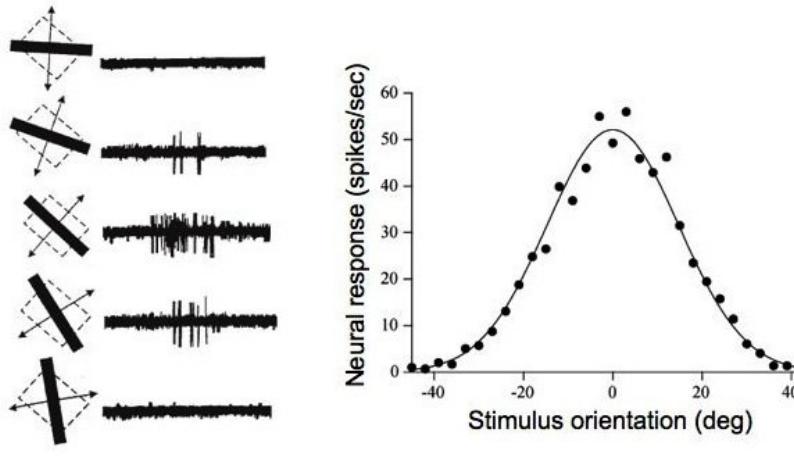


Figure 1.1: Left: Diagram of the brain from below showing the visual pathways from the retina through the LGN and to the primary visual cortex (V1) [5]. Middle: Example grating stimulus with 100% contrast at 90°. Right: Example grating stimulus with 50% contrast at 45°.



Hubel & Wiesel, 1968

Figure 1.2: An example of a tuning curve from various grating stimulus orientations collected from a monkey’s V1 to demonstrate the orientation selectivity of a V1 neuron [2].

1.2 Project Motivation and Objectives

Many recurrent models of the mouse V1 have been proposed in the past [6, 7, 8]. However, these models show conflicting connectivity rules and levels of excitation and inhibition balance. Moreover, some models were fitted to data with removed heterogeneity which was previously shown to be an important contribution to neuronal behaviour [9, 10]. On the other hand, some models were not fitted to actual V1 response data and were only simulated.

Therefore, it is currently not clear to what degree recurrent connectivity in mouse V1 contributes to orientation selectivity and how its structure gives rise to the heterogeneity of orientation tuning curves across V1 neurons.

This project aims to fill this gap by proposing a data-driven mechanistic circuit model of mouse V1. More specifically the project aims to propose a model which aligns with the following goals:

1. Accounts for heterogeneity in cortical tuning curves using randomness in model parameters, in particular the connection weights.
2. Accounts for the dependency on the types of pre-synaptic and post-synaptic neurons and their preferred orientation.
3. Allow for both loose and tight balance of excitation and inhibition or be able to interpolate between the two regimes for different values of its parameters.
4. Infer the model parameters from datasets of actual V1 tuning curves.

Chapter 2

Background

This project aims to combine 2 conflicting recurrent neural network models of the V1 in a data-driven manner. The first model supports a random untuned connectivity of neurons while the second model supports a non-random orientation-tuned connectivity. Furthermore, these 2 models also show conflict in the dynamical regime that the networks are operating in.

In this chapter, the balance of excitation and inhibition is introduced and both previous studies are thoroughly discussed and compared in terms of connectivity rule, methodology, and results.

2.1 Balance of Excitation and Inhibition

The balance of excitation and inhibition refers to the state in neural circuits where the excitatory input into neurons is counterbalanced by inhibitory input. The degree of balance could be tight, meaning the net input to a neuron is relatively small compared to individual excitatory and inhibitory components, or loose, meaning the net input to a neuron is relatively high compared to individual excitatory and inhibitory components [11]. In formal terms, the mean net input, μ , into a neuron is defined as

$$\mu = \mu_E + \mu_X - \mu_I, \quad (2.1)$$

where μ_E is the mean excitatory input, μ_X is the mean external input, and μ_I is the mean inhibitory input into a neuron. The balance of excitation and inhibition index, β , could be defined as

$$\beta = \frac{|\mu|}{\mu_E + \mu_X}. \quad (2.2)$$

If β is very small ($\beta \ll 1$) then the model operates under a tight balance while a larger β value ($\beta > 0.1$) shows that the model is operating in a loose balance. A study by Ahmadian et al. [11] suggests that the V1 likely operates in a loose balance.

Though the calculation of β is an exact method of determining the level of balance, the level of balance of excitation and inhibition could also be heuristically determined through the number of connections where a larger number of connections suggests a tighter balance while a lower number of connections often leads to a looser balance.

2.2 Previous Studies of V1

2.2.1 Random Untuned Connectivity

A study by Hansel and van Vreeswijk [6] proposed a mouse V1 model in accordance with a salt-and-pepper model where the probability of recurrent connections between neurons is purely dependent on anatomical distance.

The output response of layer 2/3 of the V1 was studied by modelling input components as three separate contributions:

1. *Recurrent Input.* The recurrent input from layer 2/3.
2. *Feed-forward Input.* The feed-forward excitatory input from layer 4.
3. *Background Input.* The background input from external neurons.

The *recurrent input* is due to the recurrent connection in layer 2/3. The network was modelled as a patch of $N_E = 40000$ excitatory and $N_I = 10000$ inhibitory neurons which were arranged randomly in a square patch of size $L \times L$ with $L \approx 1\text{mm}$. The probability of a connection between two neurons falls off with the anatomical distance according to a 2D Gaussian. This mimics the lack of a functional map in a mouse V1. Because the neurons were arranged randomly independent of the preferred orientations, the connectivity matrix, W , could be modelled as a randomly generated matrix drawn from a Bernoulli distribution,

$$W_{ij} = J_{ab}\text{Bern}(P_{ab}). \quad (2.3)$$

W_{ij} is the connection weight value from neuron index j to neuron index i . Here, J_{ab} and P_{ab} represent the synaptic efficacy and connection probability respectively between neurons of type a and b which could be excitatory, E , or inhibitory, I . An example of this weight matrix is shown in figure 2.1. Note that the red and blue dots represent the connection from excitatory neurons and the connection from inhibitory neurons respectively.

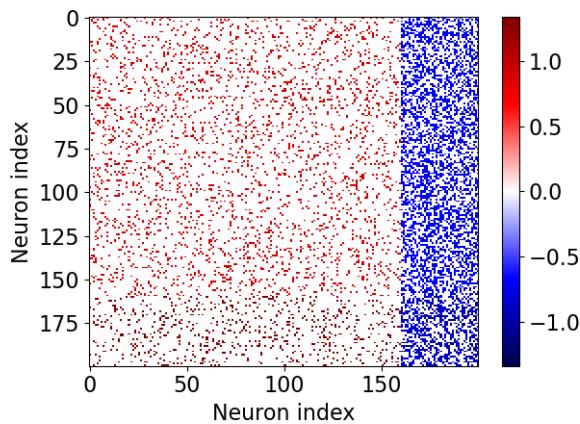


Figure 2.1: A toy example connectivity matrix from a random recurrent model with $N_E = 160$ and $N_I = 40$. The neurons are arranged in the order of type and preferred orientation.

The *feed-forward input* from layer 4 neurons. These neurons were assumed to be orientation-selective and fire in a Poisson fashion. They were randomly arranged and connected to layer 2/3 neurons depending on the anatomical distance between the neurons using a similar method to the recurrent connections. Because of this, the input from each individual feed-forward neuron was tuned to orientation but the net feed-forward input was untuned due to the random connectivity. This is depicted in figure 2.2.

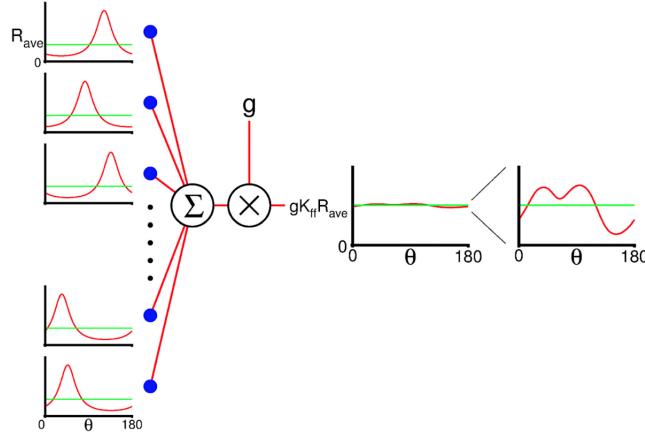


Figure 2.2: Diagram representing the tuned input from each individual feed-forward neuron and the overall untuned input into each layer 2/3 neuron [6].

The *background input* is the input contribution of many Poisson-like neurons that were not physically modelled in the network but should be considered.

This model was investigated in a non-data-driven manner and was simulated using a spiking network. From the simulation, even though the input current was not tuned, the response output of the model was tuned to orientation with heterogeneity as shown in figure 2.3. This is because even though the connection was random, the net input (the sum of the total input excitatory and inhibitory currents) was very small in contrast to the individual excitatory and inhibitory contributions. In other words, the untuned excitatory and inhibitory inputs cancelled out leaving only the tuned feed-forward excitatory input suggesting a tight balance of excitation and inhibition.

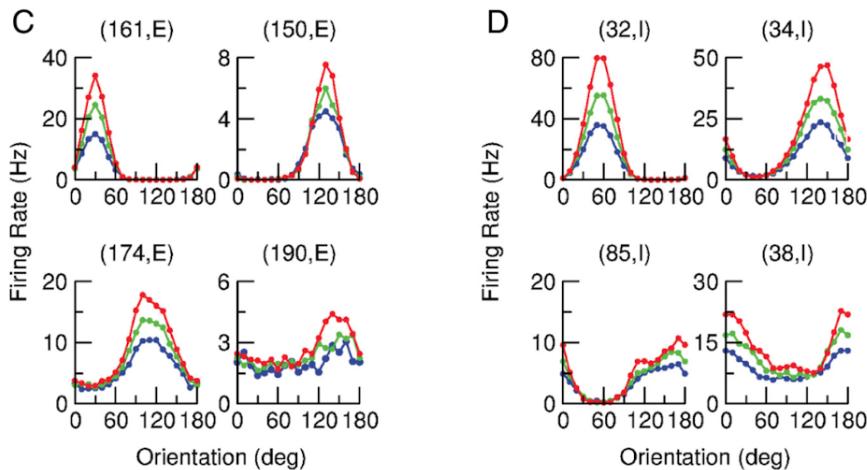


Figure 2.3: Response tuning curve from the spiking network simulation of the random recurrent model [6].

Additionally, as depicted in Figure 2.3, the tuning curves possess heterogeneity which arises from the random recurrent and feed-forward connectivity of the network. This finding aligns with subsequent recorded neural responses from experimental mouse V1 studies [7].

Furthermore, extra analysis shows that the random connectivity model also shows contrast invariance. Contrast invariance of neurons refers to the property where neurons exhibit consistent firing rates across a range of stimulus contrast values. More specifically, a neuron is said to be invariant to contrast if the response function, r , could be separated into the product of a function of orientation, f , and a function of contrast, g ;

$$r(\theta, c) = f(\theta)g(c). \quad (2.4)$$

The investigation was conducted by comparing the half-width at half-max (TW) of the response and the circular variance (CircVar) of the neuronal response under high vs medium contrast, and medium vs low contrast as shown in figure 2.4. The alignment is evidence that the model produces contrast-invariant tuning curves.

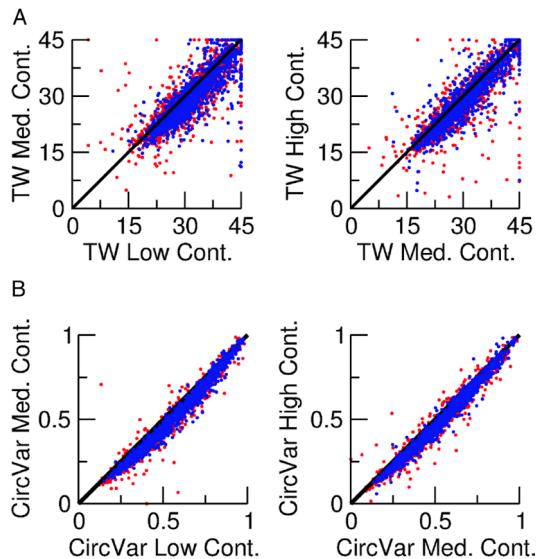


Figure 2.4: Contrast invariance shown through the alignment of the TW and CircVar values for high vs medium contrast and medium vs low contrast.

2.2.2 Non-Random Orientation-Tuned Connectivity

A data-driven study done by Kraynyukova et al [7] proposed that the recurrent connection of the mouse V1 is orientation-tuned and non-random. The neuronal response of the V1 was studied through a data-driven method. Actual V1 and LGN responses were recorded from mice *in vivo* using a grating stimulus. These responses were then tested for contrast invariance. This was analysed using the singular value decomposition (SVD) of the tuning curves as depicted in figure 2.5. Due to the fact that over 95% of the variance in the tuning curves could be accounted for by the first singular value, it was concluded that the recorded tuning curves were contrast invariant.

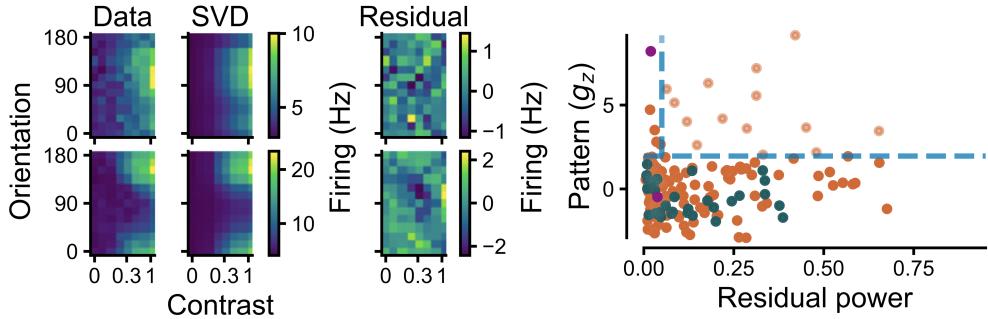


Figure 2.5: Left and Middle: The tuning curve of the data, SVD, and residual. Right: the assessment of the contrast invariance using the power of the SVD residual and the spatial autocorrelation. The blue dash lines represent the threshold value. Dark dots are contrast-invariant neurons and lighter dots are contrast-dependent neurons [7].

After verifying contrast invariance, the data was fitted with a hyperbolic ratio function and a wrapped Gaussian as shown in figure 2.6. Only the neurons whose goodness of fit exceeded a threshold were kept for further analysis. This curve fitting not only assumes contrast invariant tuning curves but also removes the heterogeneity of the tuning curves.

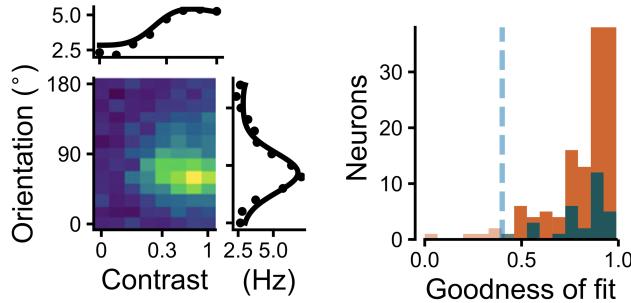


Figure 2.6: Left: The hyperbolic ratio function and Gaussian fit on the data. Right: The goodness of fit for multiple neurons. The blue dash line represents the goodness of fit threshold [7].

The augmented data outlined previously was used to fit a model where the strength of the connectivity between neurons is dependent on the preferred orientation difference. This model could be represented using a circular Gaussian of the form

$$W_{ij} = J_{ab} \exp \left\{ \frac{\cos(2(\theta_i - \theta_j)) - 1}{4\sigma_{ab}^2} \right\}, \quad (2.5)$$

where J_{ab} is the synaptic efficacy and σ_{ab} is the tuning width of recurrent connections. More precisely it is the scale over which the strength of connection falls as the difference between the preferred orientation of the pre-synaptic and post-synaptic neurons increases. $(\theta_i - \theta_j)$ is the difference in preferred orientation of neurons index i and j . An example of this weight matrix is shown in figure 2.7.

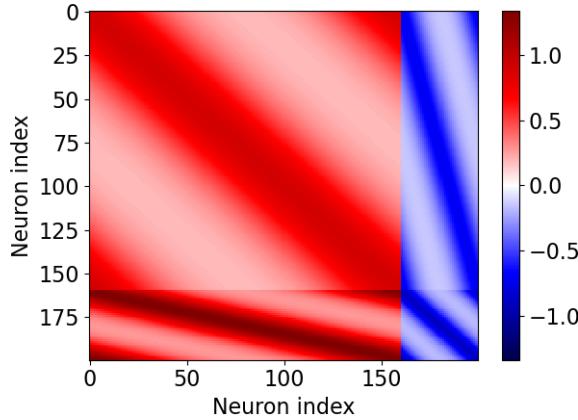


Figure 2.7: A toy example connectivity matrix from a preferred orientation dependent recurrent model with $N_E = 160$ and $N_I = 40$.

The excitatory feed-forward input connections were modelled to also be preferred orientation-dependent using the form

$$W_{ij}^{(INP)} = g_F \exp \left\{ \frac{\cos(2(\theta_i - \theta_j)) - 1}{4\sigma_F^2} \right\}, \quad (2.6)$$

where g and σ are the synaptic efficacy and tuning widths respectively and F could be either postsynaptic excitatory neurons, E , or inhibitory neurons, I .

Both the transfer function of the neurons were assumed to follow the the Stabilised Supralinear Network (SSN) model [11]. The SSN is a transfer function of the form

$$I_+^n = (\max(I, 0))^n, \quad (2.7)$$

where n is also to be inferred during optimisation. With this connectivity rule and the SSN assumption, this particular network can operate under a loose balance of excitation and inhibition [11].

To summarise, this model includes 13 parameters

- 4 Recurrent synaptic efficacy parameters, J_{ab} .
- 4 Recurrent tuning width parameters, σ_{ab} .
- 2 Feedforward synaptic efficacy parameters, $g_F^{(INP)}$.
- 2 Feedforward tuning width parameters, $\sigma_F^{(INP)}$.
- 1 SSN exponent value, n .

The contrast invariance assumption also gives the opportunity for the model to be fitted in two steps. First in contrast dimension to find efficacy values, then in orientation to find the tuning width values.

From fitting the model, it was found that there is a strong pairwise relationship between the efficacy parameters ($J_{EI} < J_{IE}$, $J_{EE} < J_{IE}$, $J_{II} < J_{IE}$, $J_{EE} < J_{II}$, $g_E < J_{EE}$, $g_E < g_I$). Furthermore, the tuning width parameters were shown to follow the order $\sigma_{EI} < \sigma_{EE} \approx \sigma_{II} < \sigma_{IE}$. The results are summarised in appendix A.1.

Chapter 3

Methods

This project proposed a new model that combines the study done by Hansel and van Vreeswijk [6] with the study by Kraynyukova et al. [7] to include both heterogeneity and preferred orientation dependency of V1 connectivity through a data-driven approach. In other words, the model proposed here is a random orientation-tuned connectivity that accounts for both loose and tight balance of excitation and inhibition.

3.1 Proposed V1 Model

3.1.1 Mean-Field Equations and The Ricciardi Transfer Function

To model the network, the leaky-integrate and fire model of the neuron was utilised [1][12]. This assumes that the neuron aggregates the past received spike train and releases a spike response when the voltage reaches the threshold voltage V_{th} at which the neuron's voltage then falls back to the reset voltage V_{reset} for a period known as the refractory period τ_{rp} .

In formal definition, the membrane potential V_i of each neuron follows the voltage law

$$\tau_i \frac{dV_i(t)}{dt} = -V_i(t) + RI_i(t), \quad (3.1)$$

where τ_i is the membrane time constant, R is the membrane resistance, and $I_i(t)$ is the input current of the neuron. Using the integrate and fire model, the current term could be modelled using the sum of the input spike train of the neuron ie.

$$RI_i(t) = \tau_i \sum_j J_{ij} \sum_k \delta(t - t_j^k - D), \quad (3.2)$$

where k is the spike index and t_j^k is the spike received at time $t - D$ where D is the synaptic delay.

Isolating one neuron from the network and then inputting a Gaussian white noise with mean μ and variance σ^2 will result in an output of the form

$$\Phi(\mu, \sigma) = \left[\tau_{rp} + \tau_A \sqrt{\pi} \int_{u_{min}}^{u_{max}} e^{u^2} (1 + erf(u)) du \right]^{-1}, \quad (3.3)$$

where $u_{min} = (V_{reset} - \mu)/\sigma$, $u_{max} = (V_{th} - \mu)/\sigma$. This equation is referred to as the Ricciardi transfer function [13]. With each neuron approximating its output in this way, the network responses could be approximated with a mean-field approximation

$$\begin{aligned} \mathbf{r} &= \Phi(\boldsymbol{\mu}, \boldsymbol{\sigma}), \\ \boldsymbol{\mu} &= W\mathbf{r} + \mathbf{h}_{ext}, \\ \boldsymbol{\sigma}^2 &= W^2\mathbf{r} + \boldsymbol{\sigma}_{ext}^2. \end{aligned} \tag{3.4}$$

The W^2 represents the element-wise square of the matrix W . \mathbf{h}_{ext} and $\boldsymbol{\sigma}_{ext}^2$ are the mean and variance of the external input stimulus respectively. The response plot of the Ricciardi transfer function at various mean and variance is shown in Figure 3.1.

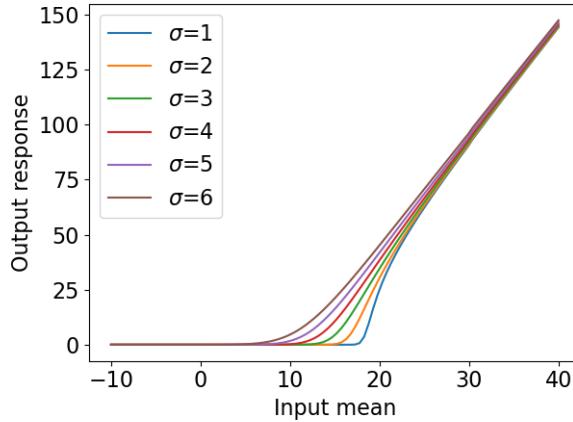


Figure 3.1: The Ricciardi transfer function for various input mean and standard deviation.

The above derivation assumes sparse connectivity, a large fixed number of presynaptic inputs from each presynaptic population, each presynaptic spike being small, all neurons are identical, and neurons fire in an asynchronous-irregular manner [1]. A detail worth mentioning is that the neurons are modelled using a mean-field approximation of the spiking network which incorporates both the mean and variance of the responses. This contrasts the study by Kraynyukova et al. [7], which only incorporates the mean in the SSN network, and the study by Hansel and Van Vreeswijk [6], which simulated the model with a spiking network. The inclusion of the variance allows the heterogeneity to be encapsulated.

In practice, Carl van Vreeswijk's implementation was adapted to approximate the Ricciardi transfer function during network simulation. Lastly, the input standard deviation, $\sigma_{ext,i}$, is assumed to be 5 for all input mean and the same for all experiments.

3.1.2 Recurrent Connectivity

To capture both the random connectivity and preferred orientation-dependent connectivity, this project's model proposes the following recurrent parameters which govern the recurrent connections in the V1:

- *Synaptic Efficacy*, J_{ab} : the strength of an existing synapse times $\sqrt{N_b}$ where N_b is the number of presynaptic neurons of type b in the network.
- *Probability of Connection*, P_{ab} : the probability of connection between neurons of the same preferred orientation.

- *Tuning Width, w_{ab} :* the tuning width of recurrent connections. More precisely the scale over which the probability of connection falls as the difference between the preferred orientation of the pre- and post-synaptic neurons increases (Note that this is similar to σ in equation (2.5) but σ in this model refers to the standard deviation of the neuronal responses rather than the tuning width).

a and b represent the type of the pre- and post-synaptic neurons respectively. This could either be excitatory, E, or inhibitory, I, and therefore results in 12 parameters.

To encapsulate the preferred orientation dependency, the probability of a connection between 2 neurons index i and j is computed using the circular Gaussian of the form

$$Z_{ij} = P_{ab} \exp \left\{ \frac{\cos(2(\theta_i - \theta_j)) - 1}{4w_{ab}^2} \right\}, \quad (3.5)$$

where ij represents a connection from the j th neuron to the i th neuron and θ is the preferred orientation of the neurons. The existence of a connection between two neurons is then determined by drawing $\{0, 1\}$ from a Bernoulli distribution with a probability Z_{ij} which is then weighted by $J_{ab}/\sqrt{N_b}$ to generate the connectivity matrix \mathbf{W} ,

$$W_{ij} = \frac{J_{ab}}{\sqrt{N_b}} \text{Bern}\{Z_{ij}\}. \quad (3.6)$$

Practically, as gradient descent was used as one of the optimisation algorithms explored it is required that backpropagation through the network and network generation should be plausible [14] (see section 3.3.1). However, the Bernoulli random variable is not differentiable. Therefore, for gradient descent, this step function behaviour is replaced with a continuous approximation

$$W_{ij} = \frac{J_{ab}}{\sqrt{N_b}} \sigma(k \cdot (Z_{ij} - X_{ij})), \quad (3.7)$$

where σ is a sigmoid function, k is a scalar hyperparameter picked to be very large, and X is drawn from a uniform distribution $[0,1]$. The weight matrix generated by this method is shown in Figure 3.2 where neurons are arranged according to their preferred orientations.

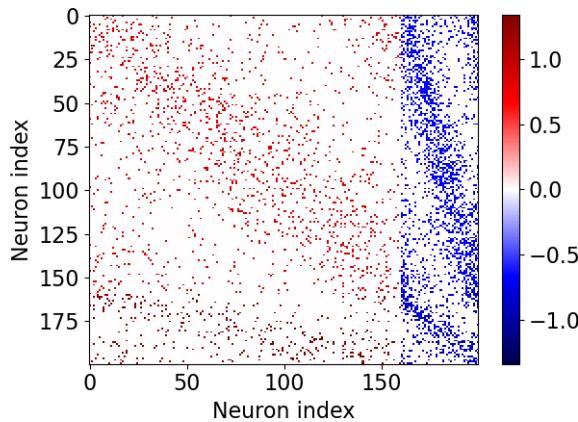


Figure 3.2: A toy example connectivity matrix from this project’s recurrent model with $N_E = 160$ and $N_I = 40$.

3.1.3 Feed-Forward Input

To simplify the model, the feed-forward input of the model was assumed to be tuned to orientation, invariant to contrast, and contains heterogeneity. The feed-forward input into a neuron index i given a grating stimulus with contrast C_{stim} and orientation θ_{stim} was assumed to follow a circular Gaussian of the form

$$h_{ext,i} = (V_{th} - V_{reset}) \times C_{stim} \times \exp \left\{ \frac{\cos(2(\theta_{stim} - \theta_i)) - 1}{4w_F^2} \right\}, \quad (3.8)$$

where w_F^2 is the width of the feed-forward response and is kept constant at 30° throughout all experiments. In order to encapsulate the heterogeneity of the feed-forward connections, the external feed-forward responses were scaled with a random variable,

$$s_{ff} = 1 + \text{Uniform}(-q_{ff}, q_{ff}), \quad (3.9)$$

where q_{ff} is another parameter to be optimised which represents the level of heterogeneity and takes the value between $[0, 1]$. It is expected that the feed-forward heterogeneity is the same across contrast for the same stimulus orientation and neuron. Therefore, s_{ff} is kept constant for the same neuron and orientation. The overall feed-forward input takes the form

$$\hat{h}_{ext,i} = s_{ff}|_{\theta,i} \times h_{ext,i}. \quad (3.10)$$

An example of this feed-forward input is shown in Figure 3.3.

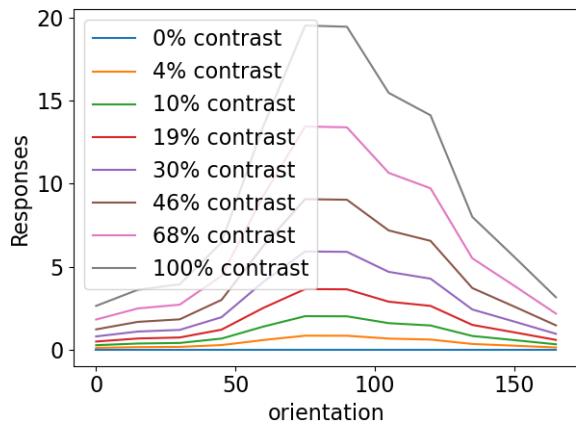


Figure 3.3: An example of the input into a neuron with a preferred orientation, $\theta_i = 90^\circ$ at various contrasts and orientations.

3.1.4 Fixed-Point Solver

To generate tuning curves, the steady state of the network has to be computed. The Euler fixed point solver [15] was utilised for this purpose where the network was executed for 300 timesteps with $\Delta t = 1ms$.

Additionally, it is desirable that the network converges to a fixed point and remains stable. Therefore, the average step of the last 20 Euler iterations across all stimuli is utilised as part of the loss function during optimisation (see section 3.2.4 for the overall loss function). The convergence is shown in appendix A.3

3.2 Loss Function and Parameter Constraints

3.2.1 Maximum Mean Discrepancy

To infer the model parameters, it is required that the statistics of the tuning curves generated by the model are as close as possible to their statistics of the dataset tuning curves. In other words, a distance metric which measures the statistical difference between two sets of tuning curves is required as a loss function during optimisation. A good candidate for this task is the Maximum Mean Discrepancy (MMD) [16], which is a distance metric that measures the mean squared difference between the statistical moments between two sets of samples from two distributions.

Given two sets of tuning curves, $X = \{\mathbf{x}_i\}_{i=1}^N$ and $Y = \{\mathbf{y}_j\}_{j=1}^M$, where each \mathbf{x} and \mathbf{y} are vectors of a tuning curve with dimensions, D , equal to the number of contrast and orientation pairs, the MMD is defined as

$$\begin{aligned} MMD^2(P, Q) &= \left\| \frac{1}{N} \sum_{i=1}^N \phi(\mathbf{x}_i) - \frac{1}{M} \sum_{j=1}^M \phi(\mathbf{y}_j) \right\|^2 \\ &= \frac{1}{N \cdot N} \sum_{i=1}^N \sum_{j=1}^N \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) - \frac{2}{M \cdot N} \sum_{i=1}^N \sum_{j=1}^M \phi(\mathbf{x}_i)^T \phi(\mathbf{y}_j) \\ &\quad + \frac{1}{M \cdot M} \sum_{i=1}^M \sum_{j=1}^M \phi(\mathbf{y}_i)^T \phi(\mathbf{y}_j) \end{aligned} \quad (3.11)$$

where ϕ is a function to match different statistical moments. Using a Gaussian kernel,

$$k(\mathbf{x}, \mathbf{y}) = \exp \left(-\frac{|\sum_{d=1}^D (x_d - y_d)|^2}{2\sigma_k} \right), \quad (3.12)$$

where x_d is each element in \mathbf{x} , the kernel trick can be applied to a universal reproducing kernel Hilbert space to get

$$\begin{aligned} MMD^2(P, Q) &= \frac{1}{N \cdot N} \sum_{i=1}^N \sum_{j=1}^N k(\mathbf{x}_i, \mathbf{x}_j) - \frac{2}{M \cdot N} \sum_{i=1}^N \sum_{j=1}^M k(\mathbf{x}_i, \mathbf{y}_j) \\ &\quad + \frac{1}{M \cdot M} \sum_{i=1}^M \sum_{j=1}^M k(\mathbf{y}_i, \mathbf{y}_j). \end{aligned} \quad (3.13)$$

From expanding the Taylor series, the MMD is equivalent to comparing infinite numbers of statistical moments [16]. As the MMD is a distance metric, the MMD has the property where $MMD = 0$ for $P = Q$ and increases the more the two distribution samples differ.

3.2.2 Tuning Curve Normalisation

As the response values at lower contrast are very low relative to values at high contrast, the MMD is not able to encapsulate the difference in two tuning curves as well and results in multiple local minima with comparable loss values during optimisation. To combat

this, the tuning curves generated are centralised to 90° and then normalised across each contrast value using the average response at that contrast. The average values of each contrast are then concatenated to form another vector, r_{avg} .

In order to calculate the difference between two sets of tuning curves, the MMD between normalised excitatory tuning curves, normalised inhibitory tuning curves, excitatory r_{avg} , and inhibitory r_{avg} are calculated independently before being summed ie.

$$\begin{aligned} MMD_{total}^2 &= MMD^2(\bar{X}_E, \bar{Y}_E) + MMD^2(\bar{X}_I, \bar{Y}_I) \\ &\quad + MMD^2(r_{avg,X_E}, r_{avg,Y_E}) + MMD^2(r_{avg,X_I}, r_{avg,Y_I}). \end{aligned} \quad (3.14)$$

$\sigma_k = 1$ was used to calculate the MMD for the normalised curves and $\sigma_k = 30$ was used to calculate the MMD for the r_{avg} vectors.

3.2.3 Assumptions and Parameter Constraints

As the 13 parameters (12 in the recurrent connections and 1 from the feed-forward input) have a known valid range, it is appropriate to constrain the parameter values to ranges. The ranges are:

- J_{ab} : 0 to 40
- P_{ab} : 0 to 0.6
- w_{ab} : 0 to 180
- q_{ff} : 0 to 1

The P_{ab} values were not constrained from 0 to 1 because the connectivity is expected to be sparse [17]. To constrain the parameters, the initial parameter values are passed through an inverse sigmoid which is scaled to the maximum value. For example, for the efficacy values,

$$\tilde{J}_{ab} = -\log\left(\frac{40}{J_{ab}} - 1\right). \quad (3.15)$$

The inverse sigmoid values, \tilde{J}_{ab} , are then the values which are optimised. For each network simulation, the parameters are converted back to the constraint parameter space using a sigmoid with the same scale.

Furthermore, from the study done by Ahmadian et al. [18], some parameters cause the network into supersaturation. Even though some supersaturation is expected in the V1 [19], preliminary testing reveals that in most optimisation, nearly all neurons fall into the supersaturation region which is undesirable. To prevent this, the pre-synaptic mean across the postsynaptic sum, ω_{ab} would need to follow a certain set of rules.

The pre-synaptic mean across the postsynaptic sum is

$$\Omega_{ab} = \frac{1}{N_a} \sum_{i \in a} \sum_{j \in b} W_{ij} = J_{ab} \frac{1}{N_a \sqrt{N_b}} \sum_{i \in a} \sum_{j \in b} Z_{ij}, \quad (3.16)$$

where Z_{ij} is defined in equation (3.5). As this calculation is expensive for large neuron populations, the following approximation was used.

Z_{ij} is a function of the difference in the preferred orientation of the neurons $f(\theta_i - \theta_j)$. The double average is therefore equivalent to performing an average once over the postsynaptic neurons; with $\theta_j = 0$

$$\Omega_{ab} = \frac{J_{ab}\sqrt{N_b}}{N_a} \sum_{i \in a} f(\theta_i). \quad (3.17)$$

The sum above could be approximated with an integral

$$\Omega_{ab} = \frac{J_{ab}\sqrt{N_b}}{2\pi} \int_{-\pi}^{\pi} f(\theta_i) d\theta_i, \quad (3.18)$$

$$\Omega_{ab} = \frac{J_{ab}P_{ab}\sqrt{N_b}}{2\pi} \int_{-\pi}^{\pi} \exp\left\{\frac{\cos(2\theta_i) - 1}{4w_{ab}^2}\right\} d\theta_i. \quad (3.19)$$

This is given by the Bessel function of the first kind which is a builtin function in PyTorch,

$$\Omega_{ab} = J_{ab}P_{ab}\sqrt{N_b} \exp\left(-\frac{1}{4w_{ab}^2}\right) I_0\left(\frac{1}{4w_{ab}^2}\right). \quad (3.20)$$

To prevent supersaturation, the following conditions will have to be met,

$$\frac{\Omega_{EE}}{\Omega_{IE}} \leq \frac{\Omega_{EI}}{\Omega_{II}} \leq 1. \quad (3.21)$$

This condition could therefore be used as a term in the lost function where the term is defined as

$$\tilde{\Omega} = \max \left\{ \max \left(\frac{\Omega_{EE}}{\Omega_{IE}} - \frac{\Omega_{EI}}{\Omega_{II}}, 0 \right), \max \left(\frac{\Omega_{EI}}{\Omega_{II}} - 1, 0 \right) \right\} \quad (3.22)$$

Moreover, the value of

$$K_{ab} = \frac{\Omega_{ab}\sqrt{N_b}}{J_{ab}}, \quad (3.23)$$

is also an alternative metric for determining the tightness of the balance of excitation and inhibition. K_{ab} represents the average number of connections between neurons of type b to neurons of type a . The larger the K_{ab} values are, the tighter the balance. Thus, the inference of the parameters would allow for the computation of K_{ab} and thus the tightness of the balance.

3.2.4 Overall Loss Function

To summarise, the overall loss function given two sets of tuning curves (the dataset of actual V1 responses and the model-generated tuning curves), as well as the current parameters used during optimisation, is

$$L = MMD_{total}^2 + \kappa_A \text{AvgStep} + \kappa_B \tilde{\Omega}, \quad (3.24)$$

where MMD_{total}^2 is defined in equation (3.14), AvgStep is the average step size across all stimuli from the last 20 Euler iterations, and $\tilde{\Omega}$ is defined in equation (3.22). κ_A and κ_B are weighting terms for the average step and supersaturation condition respectively and are set to be $\kappa_A = 0.02$ and $\kappa_B = 1$ for all the experiments.

3.3 Optimisation

3.3.1 Gradient Descent

To optimise the model parameters, the main method of optimisation is gradient descent of the form

$$\Theta \leftarrow \Theta + \eta \frac{\partial L}{\partial \Theta}, \quad (3.25)$$

where Θ is the parameter to be optimised and $\eta = 1$ is the learning rate set to a constant throughout all experiments.

The gradient of the loss with respect to the parameters is required. Therefore, the auto differentiation from PyTorch [20] was utilised to calculate the gradients using backpropagation.

From early experimentation, as PyTorch uses a dynamic computational graph when performing the forward and backward passes, the memory usage is extensive at high neuron populations. This could be combated by using forward-mode differentiation instead [21]. Using this method, a larger model of up to $N = 10000$ was able to be used for simulation. However, even though forward differentiation uses less memory, as the method requires one execution of the network per one parameter with a tangent value, the runtime is very long and unfavourable. Therefore, it is a matter of memory vs runtime situation where depending on the neuron population size and resources, forward and backward mode differentiation was utilised accordingly.

3.3.2 Exponential Natural Evolution Strategy

A method of countering the high memory usage from backward mode differentiation and the long runtime from forward mode differentiation is to optimise using another algorithm entirely. A method explored during the project is the Natural Evolution Strategy (NES) [22], in particular, the Exponential Natural Evolution Strategy (xNES).

Fundamentally, NES aims to estimate the natural gradient, $\tilde{\nabla}_\Theta L$, from parameter samples to iteratively move towards the lowest loss. This is done by using an estimation of the plain search gradient, $\nabla_\Theta L(\Theta)$, and the Fisher matrix, \mathbf{F} . The plain search gradient is given by

$$\nabla_\Theta L(\Theta) \approx \frac{1}{\lambda} \sum_{k=1}^{\lambda} f(\mathbf{z}_k) \nabla_\Theta \log(\pi(\mathbf{z}_k | \Theta)), \quad (3.26)$$

where λ is the number of parameter samples, \mathbf{z}_k , drawn from the posterior distribution $\pi(\mathbf{z} | \Theta)$ given the currently optimised parameters, Θ , at each iteration. The function f here represents the network generation, simulation, and loss calculation outlined in previous sections given the randomly drawn parameter sample. The Fisher matrix is given by,

$$\mathbf{F} = \mathbb{E} [\nabla_\Theta \log \pi(\mathbf{z} | \Theta) \nabla_\Theta \log \pi(\mathbf{z} | \Theta)^\top]. \quad (3.27)$$

The natural gradient is given by

$$\tilde{\nabla}_\Theta L = \mathbf{F}^{-1} \nabla_\Theta L(\Theta). \quad (3.28)$$

In practice, the network parameters were modelled as a multivariate Gaussian with a mean vector, $\boldsymbol{\mu}$, and covariance matrix, Σ ie. $\Theta = \{\boldsymbol{\mu}, \Sigma\}$. For each iteration of NES, both $\boldsymbol{\mu}$ and Σ are gradually updated until a stopping criterion is met.

The crucial difference between NES and xNES is the latter uses exponential parameterisation to allow Σ to always remain positive definite. The pseudo-code provided as well as the learning rates suggested by Wierstra et al. [23] was adapted to implement xNES. Furthermore, to speed up computation, important mixing was also utilised to reuse certain samples from the previous iterations [22].

Even though, xNES could be viewed as a point estimate of the parameters by taking the final $\boldsymbol{\mu}$ vector, because xNES also optimises Σ , in actuality, the outcome of xNES is a posterior distribution of the parameters which is an advantage over gradient descent.

Furthermore, as xNES does not require the use of automatic differentiation, the connectivity matrix in equation (3.6) can be used directly without the approximation in equation (3.7).

With that being said, xNES does contain many more hyperparameters to be tuned including, the sample size, λ , the learning rates for each parameter, the learning rate for the covariance matrix, and the proportion of important mixing. Hyperparameter heuristics and adaptive learning rate methods were previously proposed and were utilised in this project accordingly [23, 24].

3.4 Experimental Datasets

The model was fit to actual V1 responses. The dataset used is actual V1 responses from 92 neurons collected for the study by Kraynyukova et al. [7] recorded from a mouse in vivo. This dataset also contains the response from the varying phases and spatial frequencies of the grating stimulus. However, as the model proposed in this project only encapsulates the contrast and orientation, the responses are averaged across grating phases and spatial frequencies. An example of an excitatory tuning curve is shown in Figure 3.4.

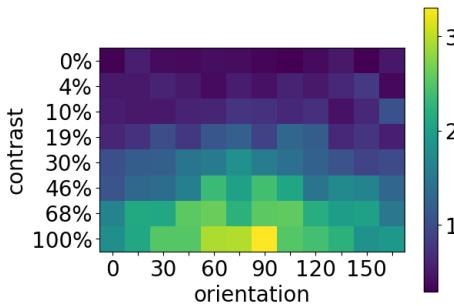


Figure 3.4: Example tuning curves from the actual mouse V1 in vivo recordings.

3.5 CNN Based Approach

The methodology thus far suggests inferring the parameter values by comparing the data tuning curves and model tuning curves using the MMD loss function and optimising model parameters. However, as will be shown in section 4, this method is not reliable due to different parameter values producing tuning curves which have a similar MMD loss to the data.

Therefore, in this project, an alternative method is proposed by using a Convolutional Neural Network (CNN) [25] to evaluate tuning curves and predict the parameter values directly. This motivation behind the CNN stems from the fact that 2D tuning curves (even without centralising) could be distinguished by the eye like an image and thus should have certain translation-invariant features for the CNN to extract. Furthermore, deeper layers in a typical CNN for image classification and regression extract features from the previous CNN layers which are also contours similar to a tuning curve [26].

The proposed CNN architecture is shown in Figure 3.5.

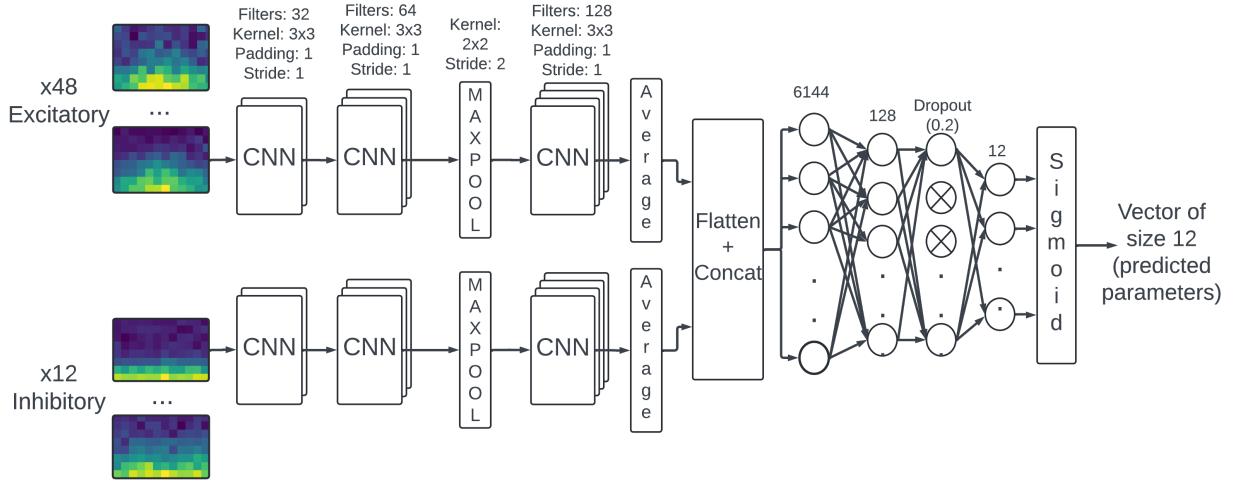


Figure 3.5: The proposed CNN architecture.

The network involves two CNN modules, one for excitatory tuning curves and one for inhibitory tuning curves. The average layer averages the output tensor of all of the input tuning curves. This allows a dynamic number of tuning curves to be inputted into the model. The outputs of the two modules are then flattened and concatenated before entering a Multilayer Perception (MLP). The final layer of the MLP is a 12-dimensional vector which is passed through a sigmoid. During training, this output was compared using the mean square error (MSE) loss with the known parameter values (scaled to the range $[0,1]$).

Even though this architecture allows a dynamic number of tuning curves, during training, 48 excitatory and 12 inhibitory tuning curves were used to train the model.

Chapter 4

Results

4.1 Method Validation

Before fitting the model to actual V1 response data, the methods must be validated to work on simulated data. The simulated data were generated from the proposed model and simulated a network using parameter values which are known to produce circuits that operate in the plausible biological regime. The values of the parameters for $N = 1000$ are shown in Figure 4.1 and will be referred to throughout the report as the “ground truth parameters”. An example tuning curve from an excitatory neuron using the ground truth parameters is shown in Figure 3.4.

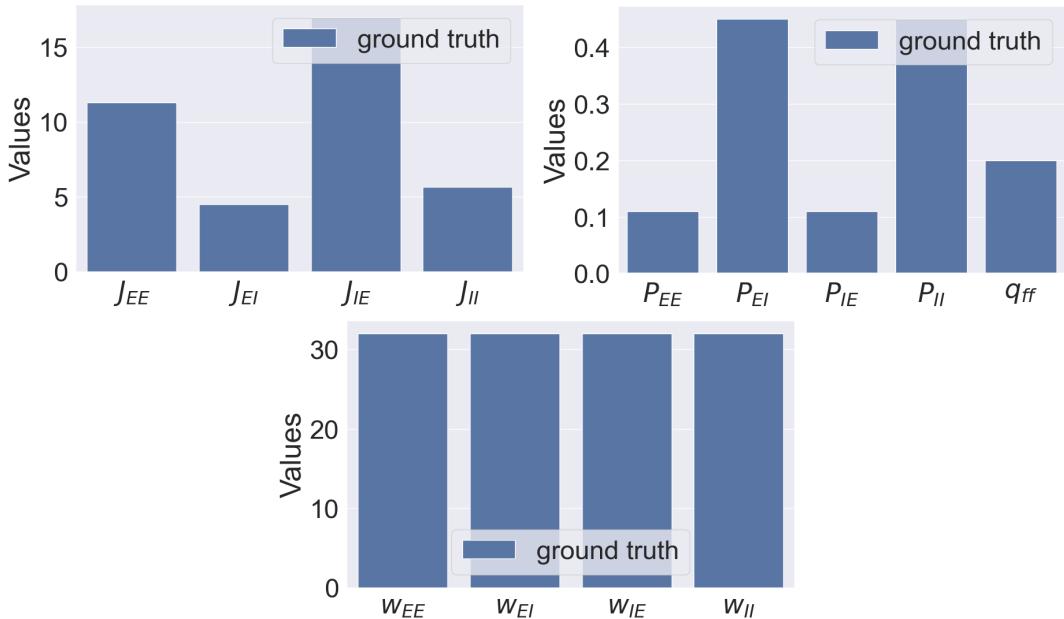


Figure 4.1: Ground truth parameter values

4.1.1 Gradient Descent with Small Neuronal Population

Firstly, to validate the method, gradient descent was utilised using the dataset of generated tuning curves from the ground truth parameters with $N_E = 800$ and $N_I = 200$. When the initial parameters were set to be exactly at ground truth, it was observed that there was negligible movement in the parameter space and the loss remained low throughout.

This is shown in Figure 4.2 and gives confidence that the method should converge to the ground truth parameters if the initial parameters are relatively close.

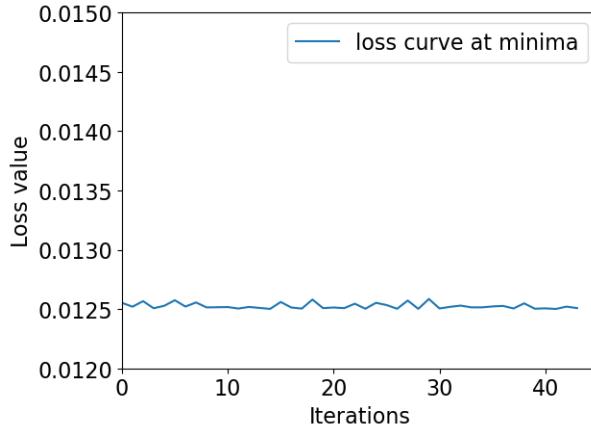


Figure 4.2: Loss from gradient descent with initial parameter values being the exact ground truth parameters used to generate the tuning curve dataset.

When the initialisation points are randomly sampled from the parameter space, it was observed that there are many local minimas. All of the different initialisations result in convergence to a different loss value and optimised parameters. Figure 4.3 shows the lowest loss and the first 2 trials from gradient descent with random initialisation. Figure 4.4 shows the lowest loss parameter values vs the ground truth values.

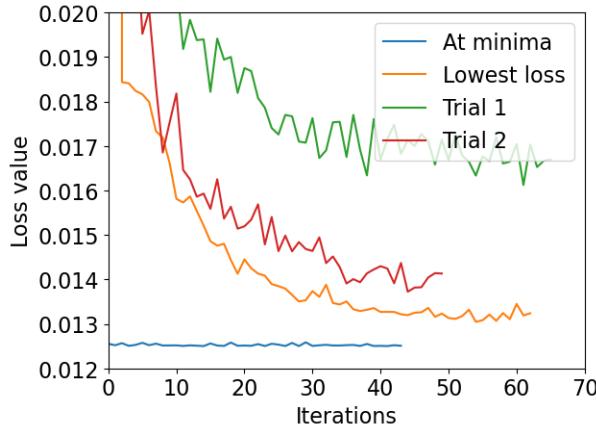


Figure 4.3: Loss from gradient descent with random initialisation. The orange line is the lowest loss value obtained. The blue line is the loss value at the ground truth parameters

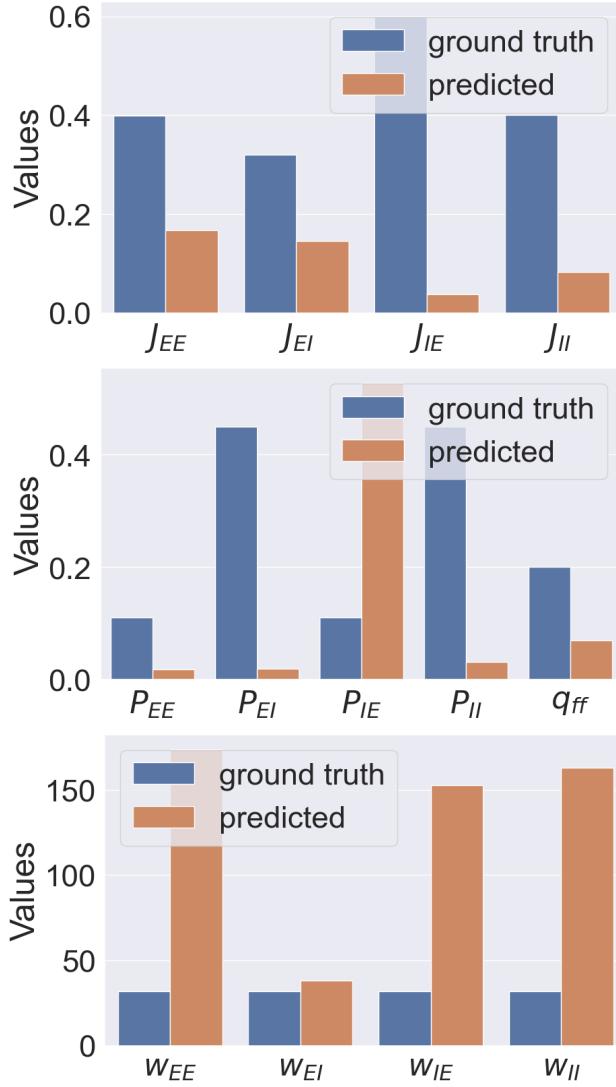


Figure 4.4: Parameter which gives the lowest loss from random initialisation vs ground truth parameters (J_{ab} are normalised with $\sqrt{N_b}$).

The results shown here demonstrate that the parameter values obtained from gradient descent at random initialisation are not able to reliably obtain the global minima.

To understand the optimised values Figure 4.5 shows the violin plot from all the minima values as well as the loss from all the gradient descent trials. From Figure 4.5, the median from optimisation does not seem to match the ground truth parameter either except for J_{EE} , J_{EI} , and P_{IE} .

An interesting speculation is that even though the data used here is a simulated dataset, the median value of the efficacy values in this specific experiment roughly follows the pairwise relationships concluded in the study by Kraynyukova et al. [7] ($J_{EI} < J_{IE}$, $J_{EE} < J_{IE}$, $J_{II} < J_{IE}$, $J_{EE} < J_{II}$).

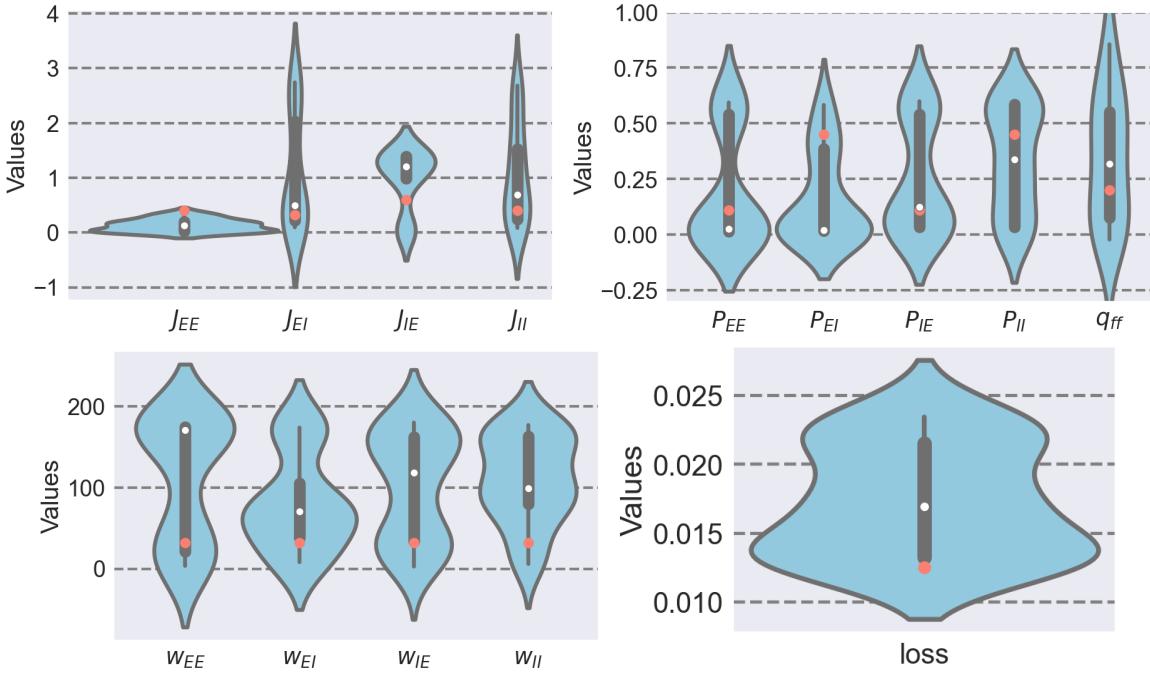


Figure 4.5: Violin plot of the final parameter values from each gradient descent trial. The white dots correspond to the median and the orange dots correspond to the ground truth values. For the loss violin plot, the orange dot corresponds to the loss at the ground truth. (J_{ab} are normalised with $\sqrt{N_b}$)

Another observation from Figure 4.5 is the fact that many predicted parameter density shows 2 distinct peaks (bi-modal) with the ground truth parameters being around the centre of one of the peaks (for example w_{IE} and P_{EE}). This suggests that 2 different parameter regimes can recreate statistically similar tuning curves to the ground truth parameters.

4.1.2 Varying sample size

For the simulated dataset, the number of tuning curves corresponds to the number of neurons, N , in the network. However, only 92 tuning curves of actual V1 neurons are available for the project. Therefore, it is important to understand the effect of reducing the number of tuning curves utilised for the MMD calculation.

To test the effect of tuning curve sample sizes, the simulated dataset size is varied from 5 to 1000 while the number of tuning curves from the network during optimisation is kept at the maximum ($N = 1000$). During all gradient descent trials, the initial parameter, as well as the random number generator seed, is kept the same for a fair comparison.

From the results, varying the sample sizes does not affect the optimised values by much. Only when the sample size is bought to 5 samples that the parameter values diverge though this is still minimal.

From further inspection, this could be because after centralising the tuning curves to 90° , the tuning curves are hardly distinguishable by qualitative means which suggests that the tuning curves are statistically similar.

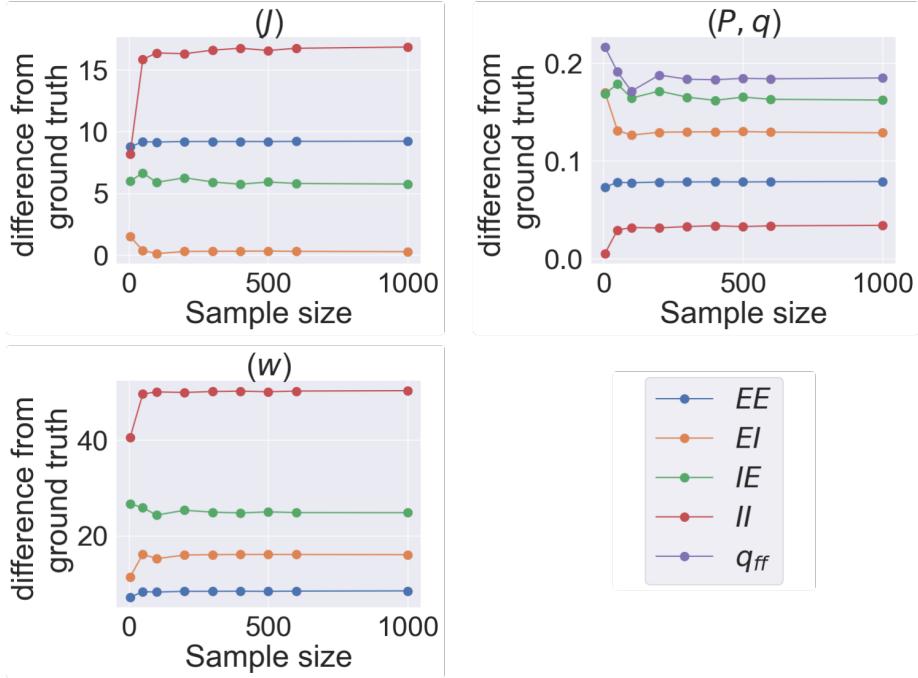


Figure 4.6: Results from varying sample sizes showing very little change at large sample sizes.

4.1.3 High Neuron Population

The previous experiments are done at low neuron population, $N = 1000$, using backwards-mode differentiation and gradient descent (BackwardGD). However, as the neurons were assumed to be asynchronous irregular, the neuron populations must remain large during simulations. However, BackwardGD fails due to high memory usage at large N . Therefore, this section aims to investigate whether xNES or forward-mode differentiation with gradient descent (ForwardGD) could be utilised to optimise for large neuron populations $N = 10000$ in place of BackwardGD.

Forward Mode Differentiation with Gradient Descent

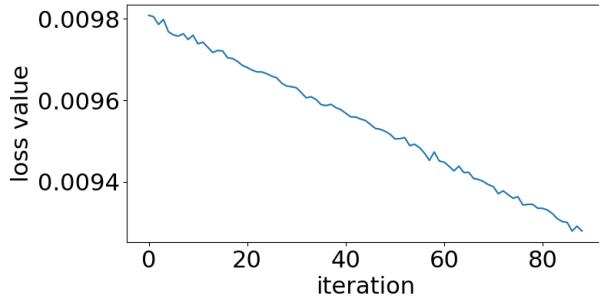


Figure 4.7: Loss from ForwardGD

ForwardGD was utilised to optimise a network with $N_E = 8000$ and $N_I = 2000$ with random initialisation. An example of a loss function is shown in Figure 4.7. However,

even though forward mode differentiation was able to optimise the model parameters, in practice only 8 valid optimisation trials can be carried out in 24 hours. This extremely long runtime could stem from the fact that currently forward mode differentiation for PyTorch is in beta and could be unoptimised compared to backward mode. The long runtime and the extremely large number of local minimas mean that ForwardGD is not yet a viable method using large neuron populations.

Exponential Natural Evolutionary Strategies

xNES was also utilised to optimise a network with $N_E = 8000$ and $N_I = 2000$ using random initialisation. Unlike ForwardGD, xNES was able to perform 30 trials in 24 hours which is more desirable. The loss curve for the trial with the lowest loss is shown in Figure 4.8. The violin plots from all the xNES trials are shown in Figure 4.9.

Again, the method of xNES using the MMD was unreliable in obtaining the ground truth value except for J_{EE} , J_{EI} , and w_{EE} . Unlike the experiment in section 4.1.1, not all the pair-wise relationships from Kraynyukova et al. [7] was observed. However, it could be speculated that the pairwise relationship may be present when more trials are used.

In many of the violin plots, bi-modal behaviour could again be observed in the parameter density with the ground truth parameters being close to the centre of each mode suggesting two parameter regimes which produce similar tuning curves.

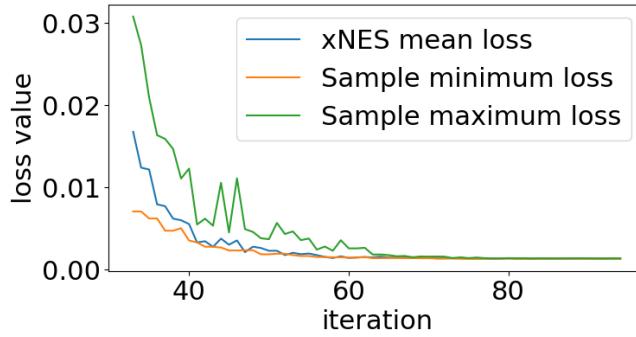


Figure 4.8: Loss from the lowest xNES trial using random parameter initialisation. The green line represents the maximum loss of the samples during each xNES iteration. The orange line is the minimum loss. The Blue line is the loss of the mean vector, μ , which are the parameter values.

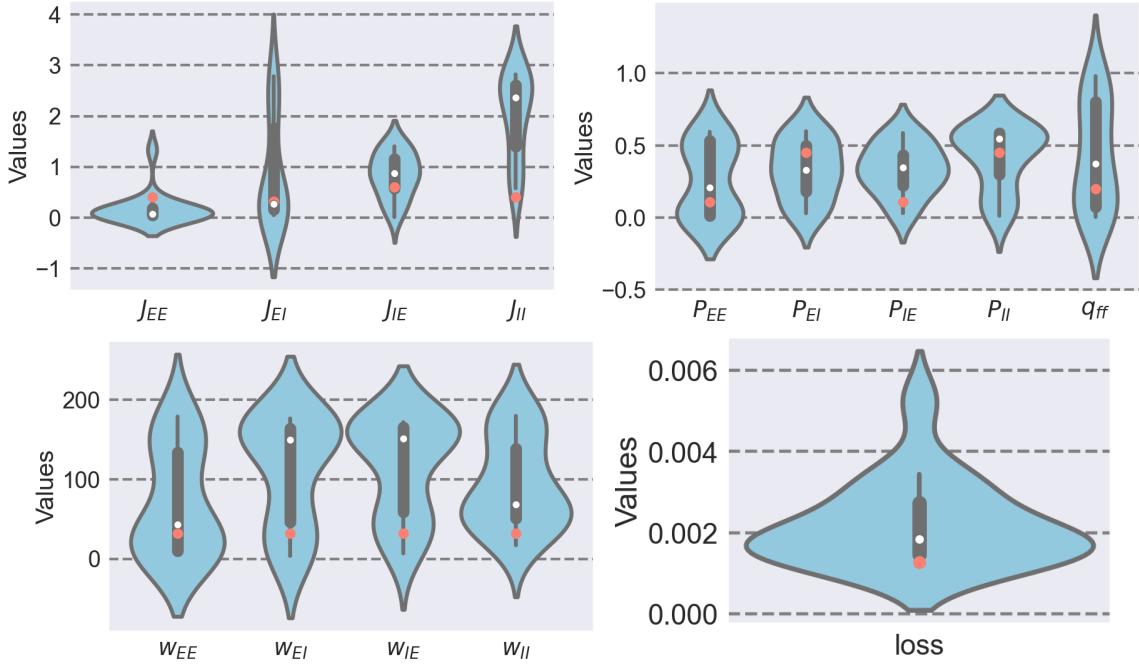


Figure 4.9: Violin plot of the final parameter values from each xNES trial. The white dots correspond to the median and the orange dots correspond to the ground truth values. For the loss violin plot, the orange dot corresponds to the loss at the ground truth. (J_{ab} are normalised with $\sqrt{N_b}$)

4.2 Fitting Model to Actual V1 Responses

Even though, the method validation suggests that the proposed method was not able to consistently recover the ground truth parameters, it is still valuable to see what parameters were recovered using real data.

As ForwardGD proves impractical due to the long runtime, BackwardGD at $N = 1000$ and xNES at $N = 10000$ were utilised on actual data outlined in section 3.4.

Figure 4.10 shows the violin plot from the experiment with BackwardGD and Figure 4.11 shows the same plot for xNES. Again, two distinct peaks are present in many parameter densities from both experiments. Furthermore, the connection to excitatory neurons seems sparser than the connection to inhibitory neurons. This could be concluded from the lower P median values for the connection with a postsynaptic excitatory neuron.

A distinction between smaller and larger networks could also be observed, in a smaller network $N = 1000$, it is observed that the feed-forward heterogeneity, q_{ff} , is consistently high compared to that of a larger network. Furthermore, the tuning width in the smaller network seems wider than that of the larger network.

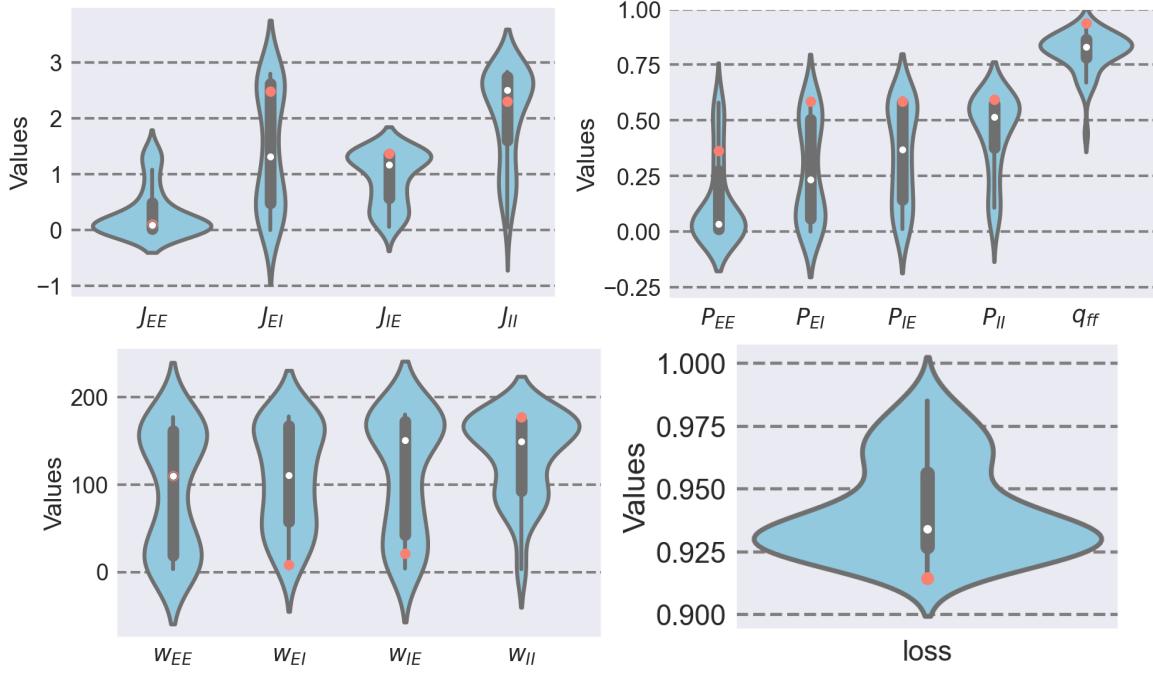


Figure 4.10: Violin plot of the final parameter values from each BackwardGD trial with $N = 1000$. The white dots correspond to the median and the orange dots correspond to the lowest loss values. (J_{ab} are normalised with $\sqrt{N_b}$)

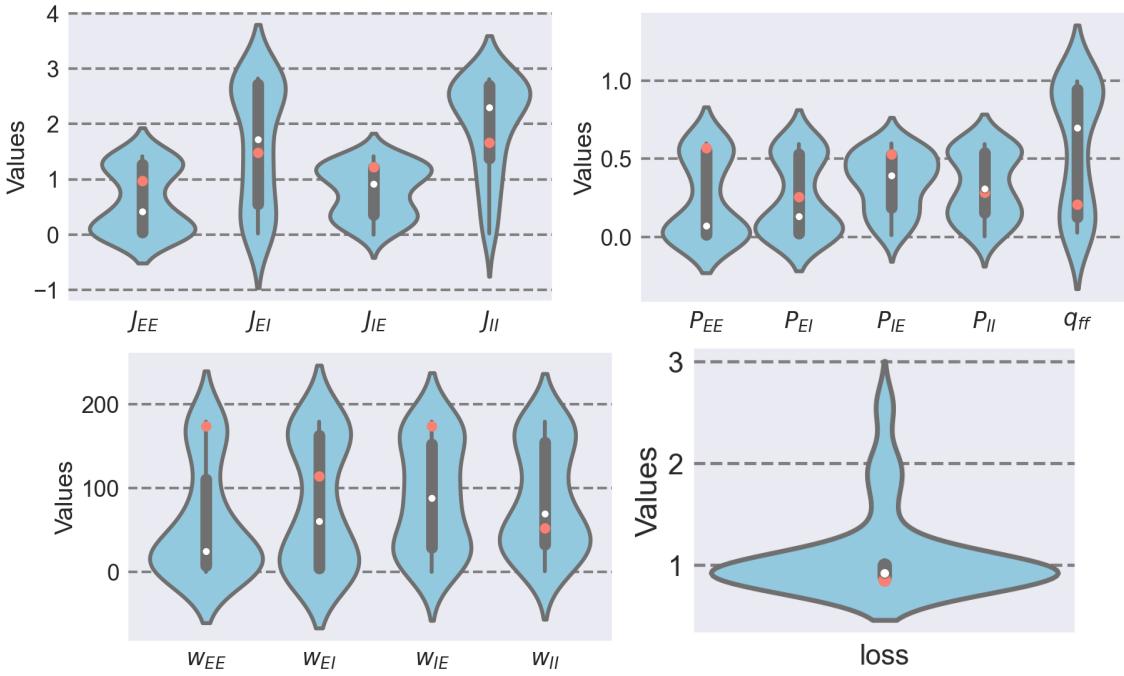


Figure 4.11: Violin plot of the final parameter values from each xNES trial with $N = 10000$. The white dots correspond to the median and the orange dots correspond to the lowest loss values. (J_{ab} are normalised with $\sqrt{N_b}$)

To investigate the contrast invariance of the network, the point estimate was taken to be the lowest lost value from the xNES optimisation. The SVD was performed on the tuning curves and the first singular value magnitude was compared to the singular value

vector magnitude. As most of the data could be explained by the 1st singular value, the model was concluded to be contrast invariance. This is shown in Figure 4.12 where the histogram of both the data and model tuning curves are shown and are very similar.

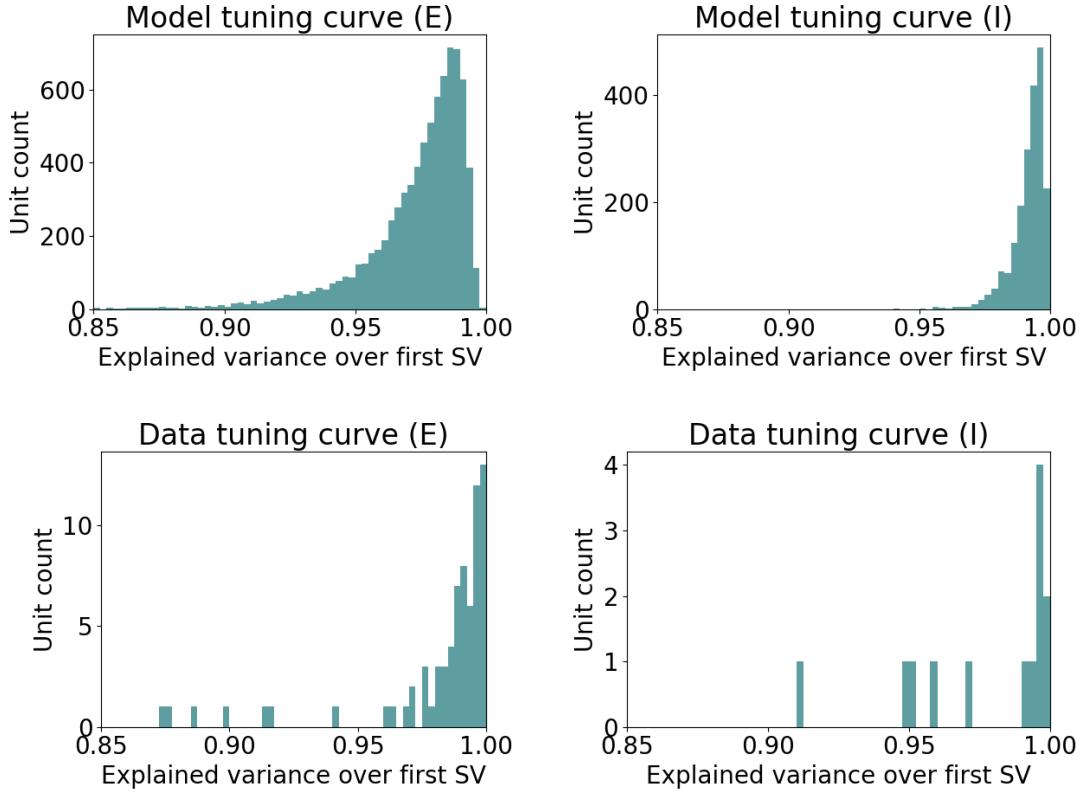


Figure 4.12: Histogram of the percentage of the magnitude of the first singular value compared with the magnitude of the singular value vector. Top Right: Model excitatory. Top Left: Model inhibitory. Bottom Right: Data excitatory. Bottom Left: Data inhibitory

To explore the model's balance of excitation and inhibition, the value of K_{ab} from equation (3.23) which represents the average number of received connections between neurons of type b to neurons of type a was plotted on a violin plot in Figure 4.13. The value of K_{ab} as a heuristic which can give insights into the tightness of the balance of excitation and inhibition with higher K_{ab} suggesting a tight balance.

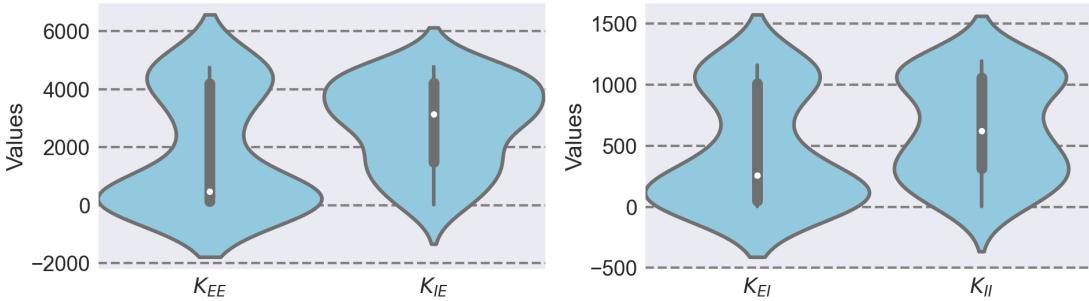


Figure 4.13: The violin plot of the average number of connections of each type.

The higher the value of K , the tighter the balance. From the figure above, there are again two modes across the densities. This suggests that our model can operate in both loose and tight balance with local minima values being in either loose or tight. Both loose and tight balances can produce tuning curves which are statistically similar to that of the data depending on the initial conditions hence both loose and tight balance models are valid. However, because the V1 of rodents biologically lack a functional map unlike cats and monkeys [4], it could be speculated that a more random connection with a tighter balance, similar to that of Hansel and van Vreeswijk's proposed model [6], is what is present in the mouse V1.

4.3 CNN Preliminary Results

As the results shown in previous experiments point towards the methods explored so far being unable to recover the ground truth parameters due to many local minimas, a different method was explored. Precisely, the CNN proposed in section 3.5 was trained and evaluated. To simplify the inference task, the feed-forward inputs were modelled without heterogeneity; equation (3.8).

First, a dataset of 10000 sets of tuning curves was collected by simulating the network of size $N = 1000$ at random parameter values. Only the parameters which follow equation (3.21) were used. Each set of tuning curves contains 48 excitatory and 12 inhibitory tuning curves to reflect the actual ratio in the network. It should however be mentioned that the proposed model can handle any number of tuning curves due to the averaging step after the two CNN modules.

The collected dataset was then used to train the network with a batch size of 32 using the Adam optimiser. A validation set was utilised to prevent overfitting. The loss curve is shown in Figure 4.14.

The model was able to achieve a testing MSE of 0.0823. However, upon plotting the predicted vs actual plot for the 12 parameters, it is observed that the model performance was still poor. Many predictions were made around a small range eg. w_{EI} , J_{EE} . However, the inference of parameters such as J_{IE} , P_{EI} and w_{IE} , though they perform poorly, show a promising trend.

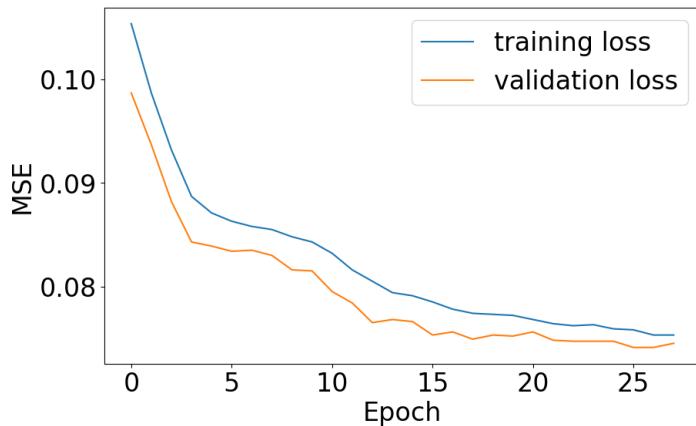


Figure 4.14: The training and validation loss from the CNN model.

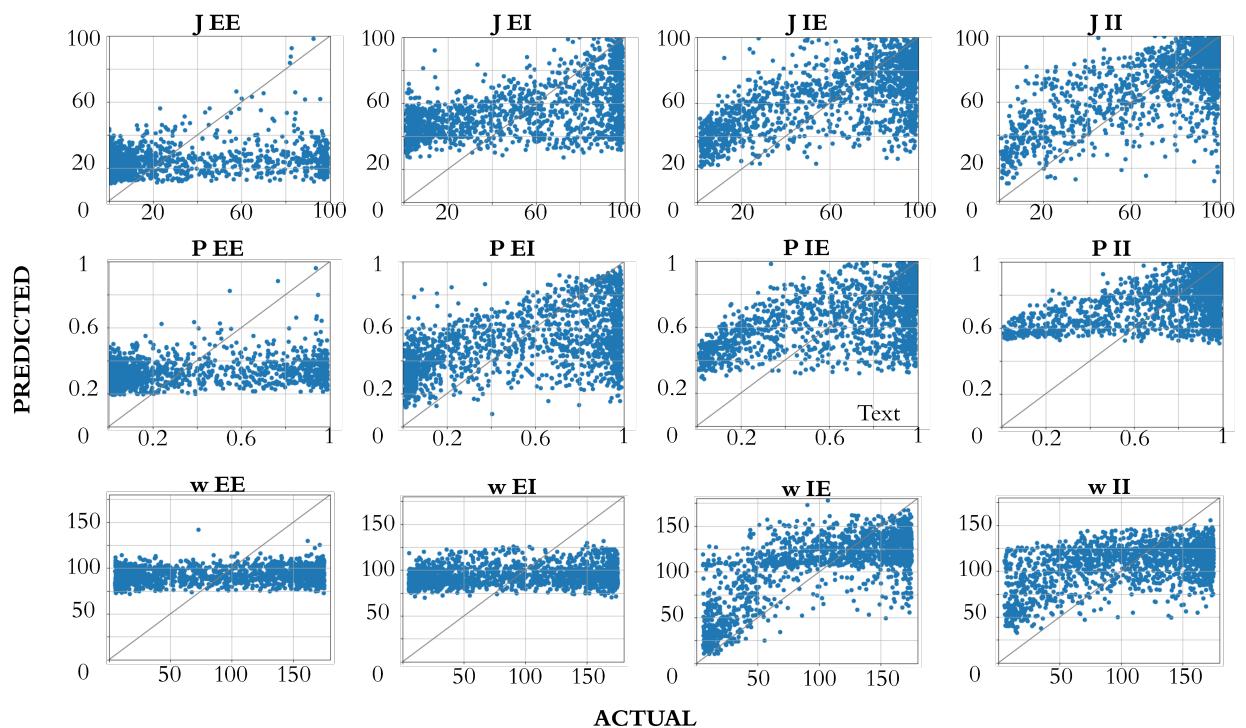


Figure 4.15: The Predicted vs Actual plot for all 12 parameters

Chapter 5

Conclusion and Future Directions

5.1 Conclusion

In previous studies of the V1, many conflicting theories have arisen. The conflicts include different connectivity rules as well as a different balance of excitation and inhibition levels. Therefore, it is not clear to what extent the recurrent connectivity gives rise to orientation selectivity and heterogeneity across tuning curves.

In this study, a new mouse V1 model with a random orientation-tuned recurrent connectivity was proposed to explain the orientation selectivity and heterogeneity across neurons. The model was allowed to operate in both a tight and loose balance of excitation and inhibition.

Furthermore, to fit the model in a data-driven manner, various methodologies were explored in detail including the xNES and gradient descent using backward differentiation or forward differentiation. It could be concluded that at lower neuron populations, gradient descent with backward differentiation is a viable tool to fit the model to data. Meanwhile, at higher neuron populations, xNES is superior due to less memory usage.

From fitting our model to both simulated and actual data, many local minimas are present. Furthermore, two distinct modes of parameter density arise which produce tuning curves that are statistically similar to the data. Moreover, the balance of excitation and inhibition also possesses a bimodal property; loose and tight balance. This suggests that both loose and tight balances are able to produce heterogeneous tuning curves that are statically similar to that of the dataset. From this, it could be speculated that both the model proposed by Hansel and van Vreeswijk and Kraynyukova et al. are special cases of this project's model. However, further experimentation would have to be performed to confirm this hypothesis.

From the results, another hypothesis is that in animals such as rodents, where the V1 are arranged in a salt-and-pepper structure, the V1 operates in a tighter balance with random weak-orientation-tuned connectivity while in other animals with a functional map, the V1 operates in a looser balance with random strong-orientation-tuned connectivity. This random connectivity could also explain the high heterogeneity across mouse V1 neurons compared to that of other species with a functional map.

In the last section of this study, a CNN-based model was proposed to estimate the parameters directly from tuning curves. Though the preliminary results are still poor, some promising trends could be seen. The sub-optimal results could stem from many problems. For instance, the CNN might not be extracting any translation invariance features [27] or the tuning curves do not contain any translation invariance features in

the first place. Moreover, the averaging layer could be obscuring valuable information from the CNN layer. Lastly, the sigmoid layer could cause gradient vanishing issues in the network and should be investigated further [28].

It should also be mentioned here that the model was not yet biased with biological knowledge. Therefore, the predicted model parameters might not fall into a biologically valid regime. It is very likely that with an introduction of this bias into the loss function, the model would be able to perform more accurately. An alternative would also be to use the CNN prediction as an initialisation point for xNES or gradient descent with the MMD.

5.2 Future Directions

In the future, the feed-forward connectivity of the model proposed here could be improved to be similar to that of the recurrent connectivity with random orientation-tuned connectivity instead of the model in section 3.1.3. This is so that the model could be better compared with the study by Kraynyukova et al. [7] directly. Moreover, the stimulus could be extended to include the spatial frequency and phase as this is available in the dataset utilised. Furthermore, the model should be fitted to other V1 datasets from both mice and other animals with and without a functional map to reproduce the results and confirm the hypotheses inferred from the results.

In terms of the deep-learning approach, many more experiments and improvements should be performed including but not limited to:

- Validating the use of the CNN by exploring each CNN layer output contours as well as replacing the CNN with an MLP layer instead and comparing the prediction performance.
- Validate the average layer by exchanging the average layer with a concatenation layer and removing the dynamic number of tuning curves features.
- Explore other state-of-the-art model architecture. This could include, the transformer encoder model without positional encoding where each input vector is the normalised tuning curve.
- Perform hyperparameter optimisation using Bayesian optimisation. Perhaps using tools such as Optuna [29].
- Expand the model to output a posterior distribution of the parameter rather than a point estimate. The posterior distribution would allow a better analysis of the model and allow methods like xNES a better distribution assumption than that of a multivariate Gaussian.

Lastly, overall, this study lays the foundation for more comprehensive models of the V1 by bridging the gap between two conflicting models and proposing new methods for future research and improvement. Further validations will be crucial in solidifying the model’s accuracy as well as method applicability in the field. These studies would be essential to explaining the heterogeneity, balance of excitation and inhibition, and the underlying mechanism of the connectivity in the V1.

Repository

The code used in this project can be found at <https://github.com/paopaoch/Mouse-V1>.

Bibliography

- [1] Peter Dayan and L. F. Abbott. *Theoretical Neuroscience : Computational and Mathematical Modeling of Neural Systems*. The MIT Press, 2001.
- [2] Hubel DH. and Wiesel TN. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J Physiol*, 160(1):106–54, January 1962.
- [3] Ian Antón Oldenburg et al. The logic of recurrent circuits in the primary visual cortex. *Nature Neuroscience*, 27(1):137–147, January 2024.
- [4] Jaeson Jang et al. Retino-cortical mapping ratio predicts columnar and salt-and-pepper organization in mammalian visual cortex. *Cell Reports*, 30(10), March 2020.
- [5] Paul C. Bressloff. Course 11 pattern formation in visual cortex. *Les Houches*, pages 477–574, 2005.
- [6] David Hansel and Carl van Vreeswijk. The mechanism of orientation selectivity in primary visual cortex without a functional map. *The Journal of Neuroscience*, 32(12):4049–4064, March 2012.
- [7] Nataliya Kraynyukova et al. The mechanism of orientation selectivity in primary visual cortex without a functional map. *Proceedings of the National Academy of Sciences*, 119(41), August 2022.
- [8] Jason M. Samonds, Brian R. Potetz, Christopher W. Tyler, and Tai Sing Lee. Recurrent connectivity can account for the dynamics of disparity processing in v1. *The Journal of Neuroscience*, 33(7):2934–2946, Feb 2013.
- [9] David Hansel and Carl van Vreeswijk. How noise contributes to contrast invariance of orientation tuning in cat visual cortex architecture in the cat's visual cortex. *The Journal of Neuroscience*, 22(12):5118–5128, June 2002.
- [10] Jorrit S Montijn, Pieter M Goltstein, and Cyriel MA Pennartz. Mouse v1 population correlates of visual detection rely on heterogeneity within neuronal response patterns. *eLife*, 4:e10163, dec 2015.
- [11] Yashar Ahmadian and Kenneth D. Miller. What is the dynamical regime of cerebral cortex? *Neuron*, 109(21):3373–3391, November 2021.
- [12] Alessandro Sanzeni et al. Response nonlinearities in networks of spiking neurons. *PLOS Computational Biology*, 16(9), September 2020.
- [13] Luigi M. Ricciardi. *Diffusion processes and related topics in biology*. Springer-Verlag Inc, Berlin; New York, 1977.

- [14] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
- [15] Steven C. Chapra and Raymond P. Canale. *Numerical Methods for Engineers*. McGraw-Hill Education, 2015.
- [16] Yujia Li et al. Generative moment matching networks. *Preliminary work on arXiv*, February 2015.
- [17] Stephanie C Seeman et al. Sparse recurrent excitatory connectivity in the microcircuit of the adult mouse and human cortex. *eLife*, 7:e37349, sep 2018.
- [18] Yashar Ahmadian et al. Analysis of the stabilized supralinear network. *Neural Computation*, 25(8):1994–2037, July 2013.
- [19] Jonathan W. Peirce. The potential importance of saturating and supersaturating contrast response functions in visual cortex. *Journal of Vision*, 7(6), Apr 2007.
- [20] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- [21] Andreas Griewank and Andrea Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2nd edition, 2008.
- [22] Daan Wierstra et al. Natural evolution strategies. *arXiv*, June 2011.
- [23] Tobias Glasmachers et al. Exponential natural evolution strategies. *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, 2010.
- [24] Isao Ono Masahiro Nomura. Towards a principled learning rate adaptation for natural evolution strategies. *EvoApplications2022*, 2022.
- [25] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, 2016.
- [26] Chaohui Wang, Huan Fu, Dacheng Tao, and Michael Black. Occlusion boundary: A formal definition its detection via deep exploration of context. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 11 2020.
- [27] Valerio Biscione and Jeffrey S. Bowers. Convolutional neural networks are not invariant to translation, but they can learn to be. *Journal of Machine Learning Research*, 22:1–28, Sep 2021. Submitted 1/21; Revised 7/21; Published 9/21.
- [28] Leni Ven and Johannes Lederer. Regularization and reparameterization avoid vanishing gradients in sigmoid-type networks. *arXiv preprint arXiv:2106.02260*, 2021. arXiv:2106.02260 [cs.LG].
- [29] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. page 2623–2631, 2019.

Appendix A

Supporting Results

A.1 Results From a Non-Random Orientation-Tuned Connectivity

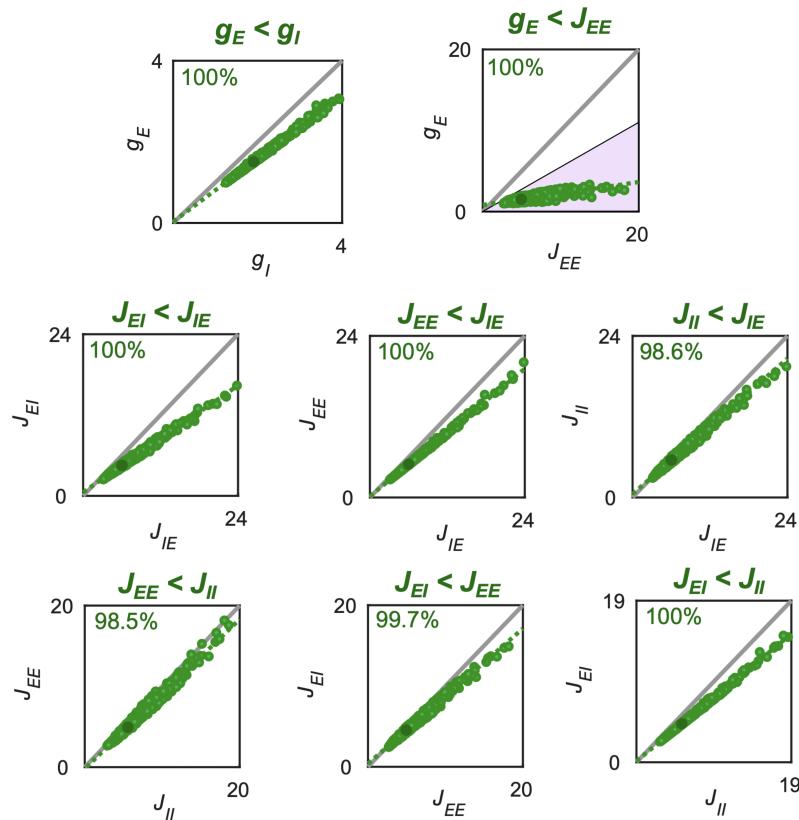


Figure A.1: Synaptic efficacy relationships from the study by Kraynyukova et al. [7] for the non-random orientation-tuned network.

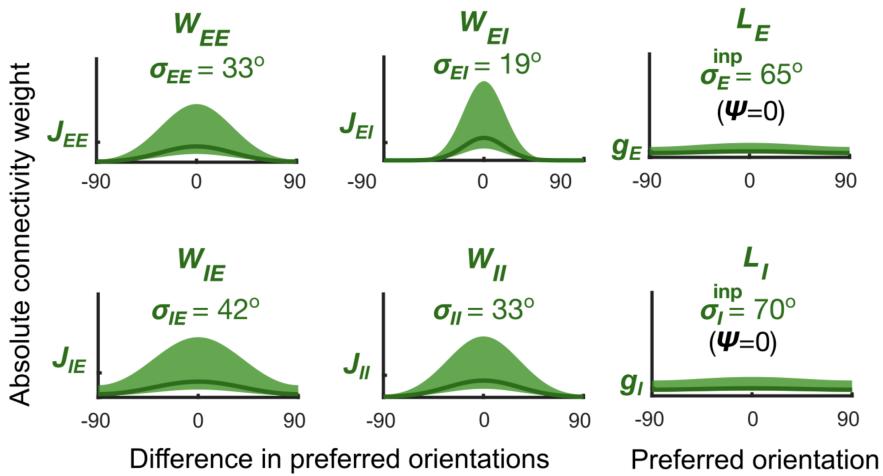


Figure A.2: Tuning width relationships from the study by Kraynyukova et al. [7] for the non-random orientation-tuned network.

A.2 Tuning Curves Examples

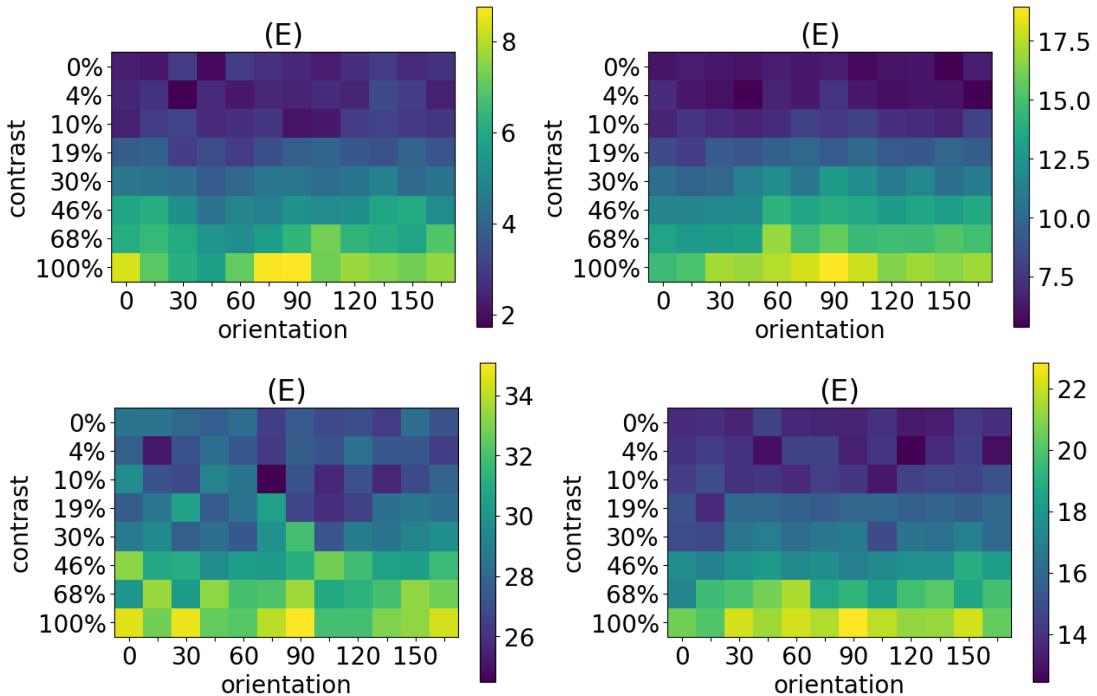


Figure A.3: Example excitatory tuning curves from the dataset [7].

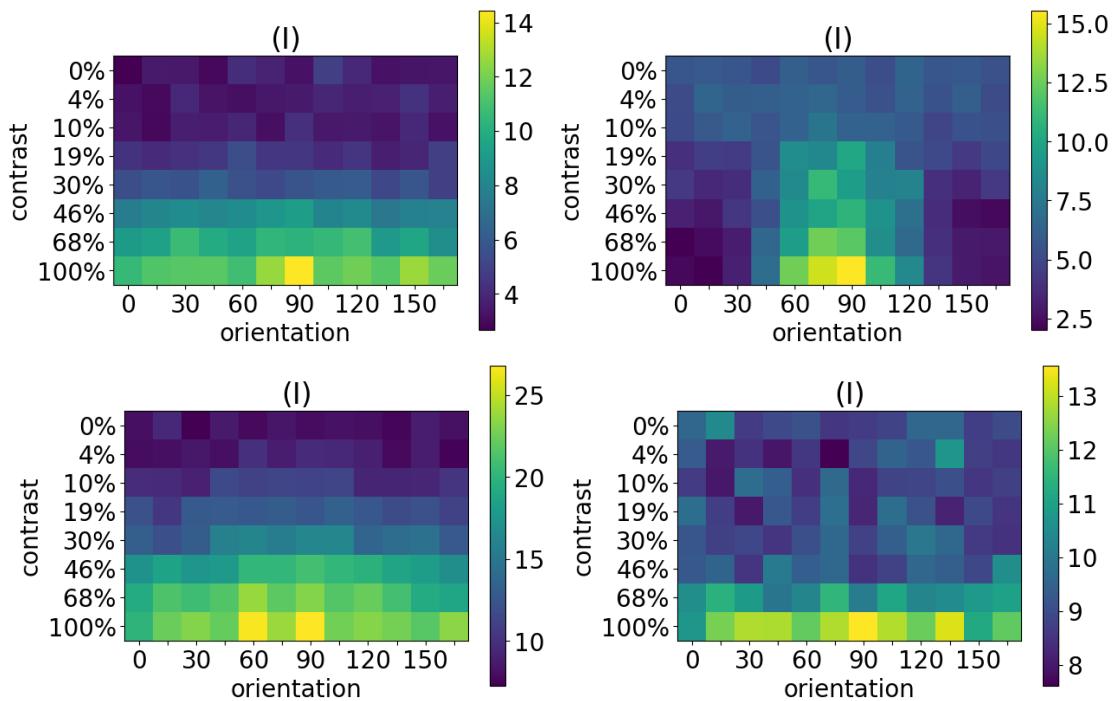


Figure A.4: Example inhibitory tuning curves from the dataset [7].

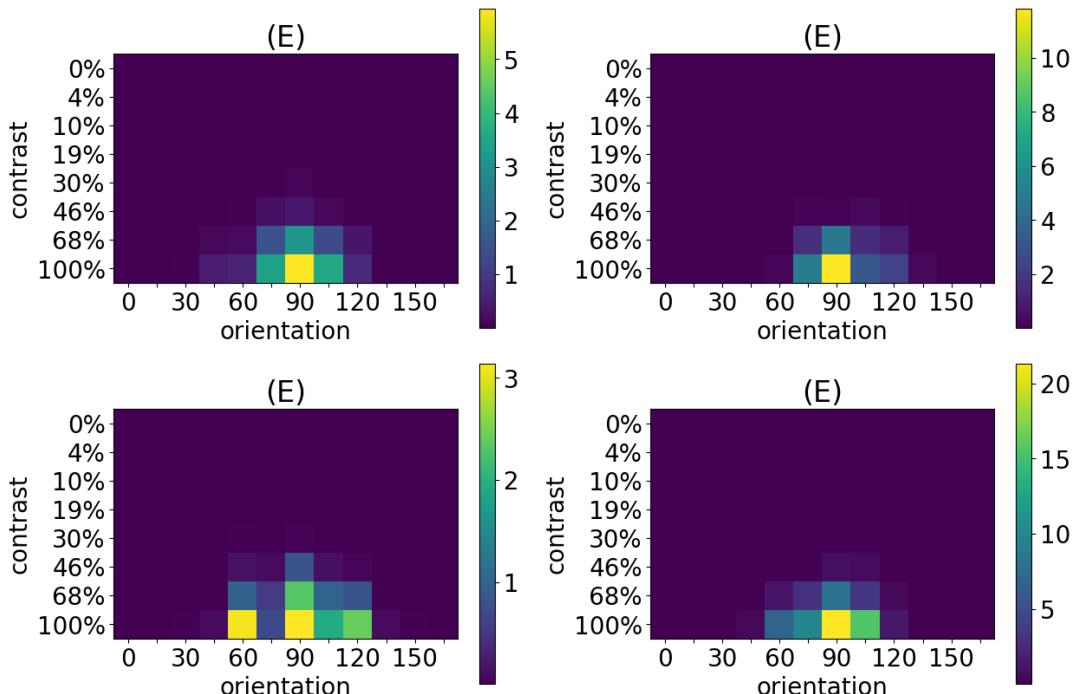


Figure A.5: Example excitatory tuning curves from ground truth parameters for method validation at $N = 10000$.

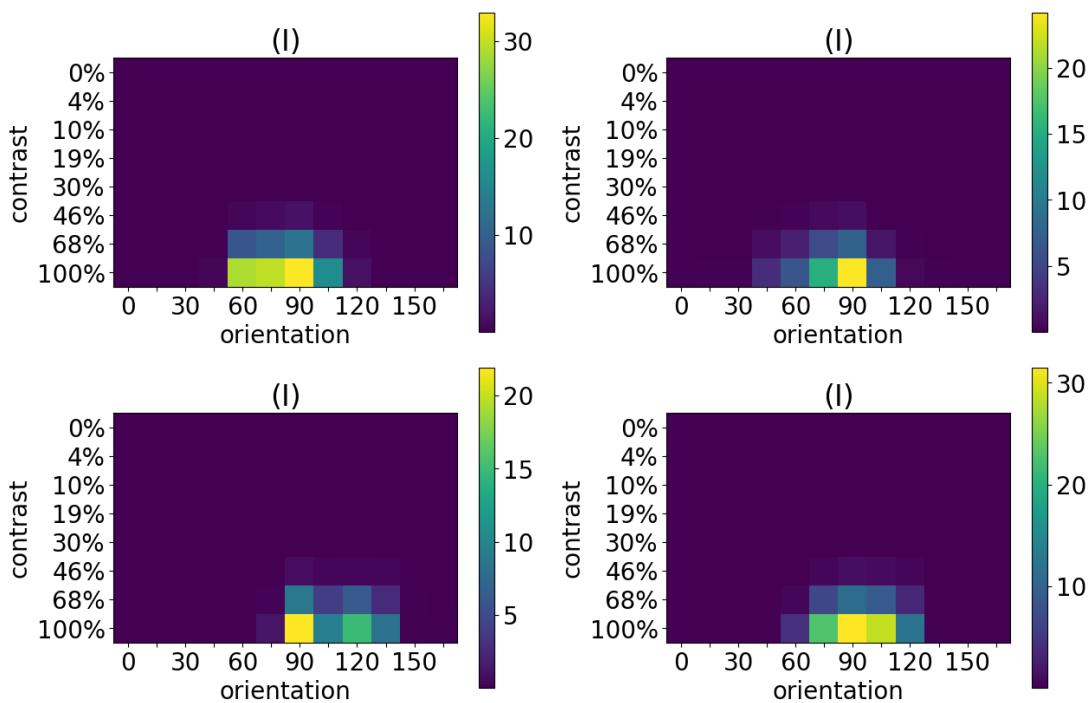


Figure A.6: Example inhibitory tuning curves from ground truth parameters for method validation at $N = 10000$.

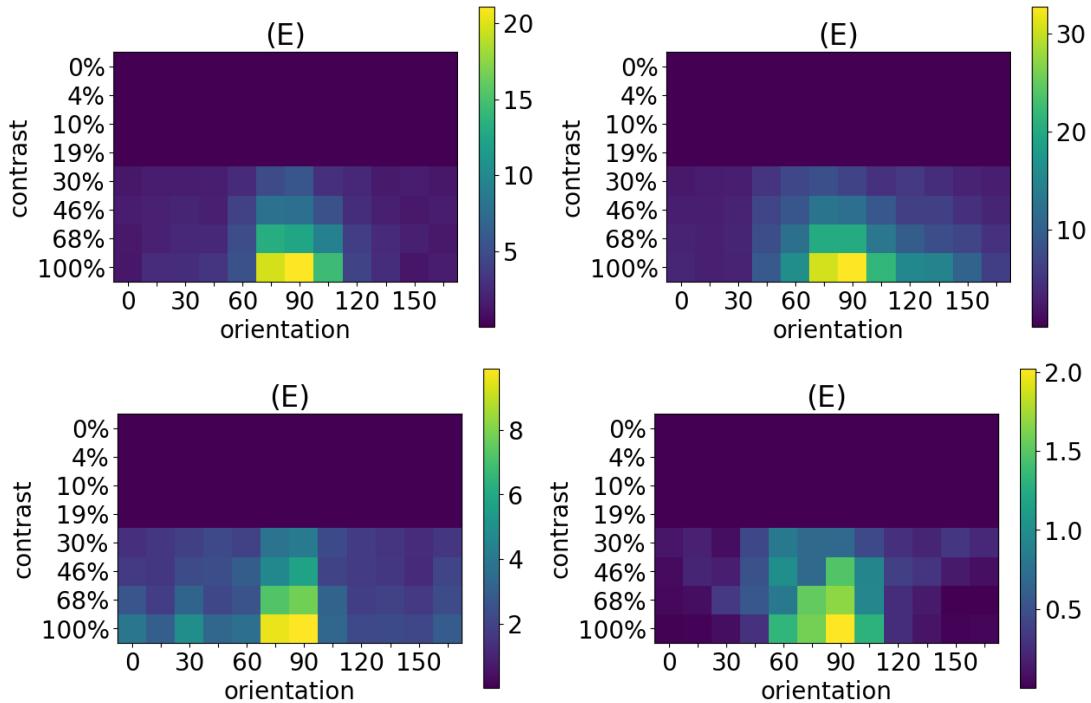


Figure A.7: Example excitatory tuning curves from xNES optimisation with real data at $N = 10000$.

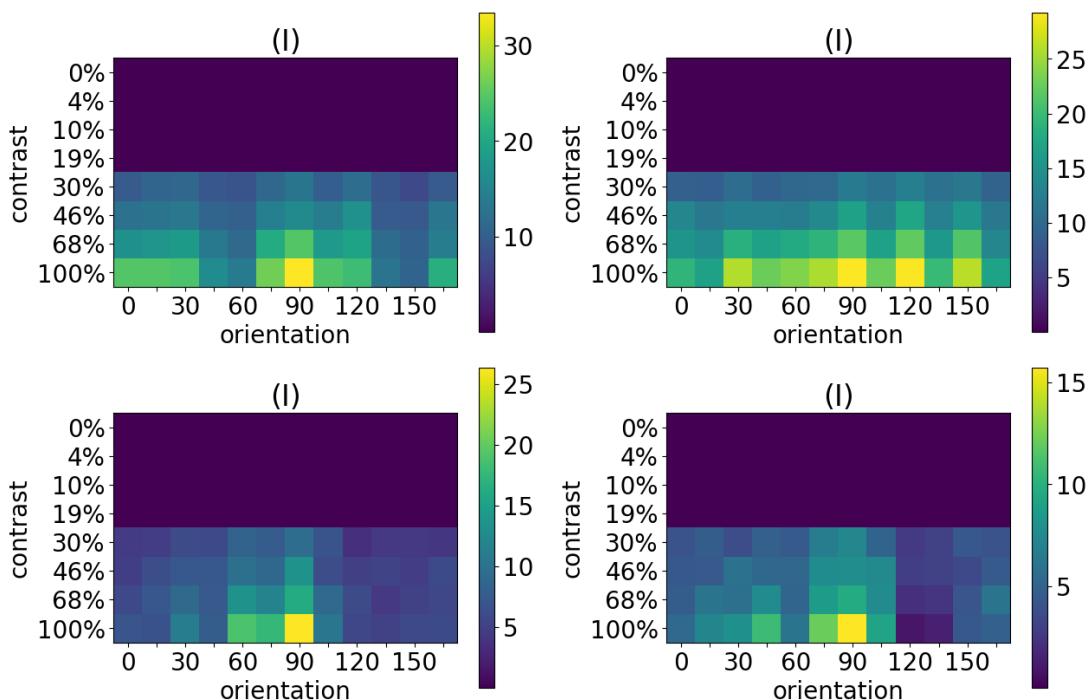


Figure A.8: Example inhibitory tuning curves from xNES optimisation with real data at $N = 10000$.

A.3 Neuronal Activity

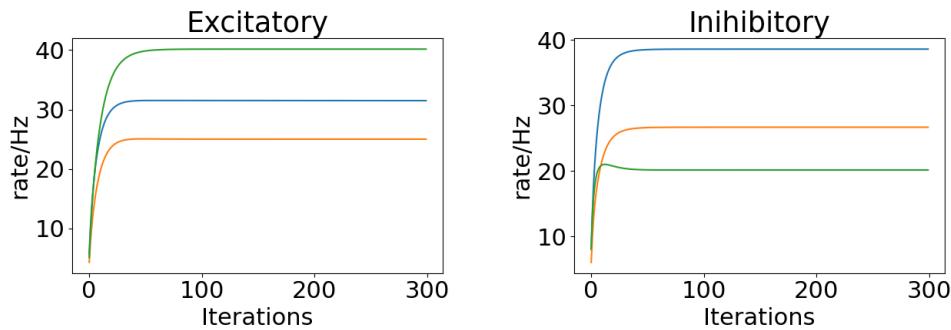


Figure A.9: The example neuronal response over time (from the Euler fixed-point solver) for excitatory and inhibitory neurons using the model proposed during method validation.

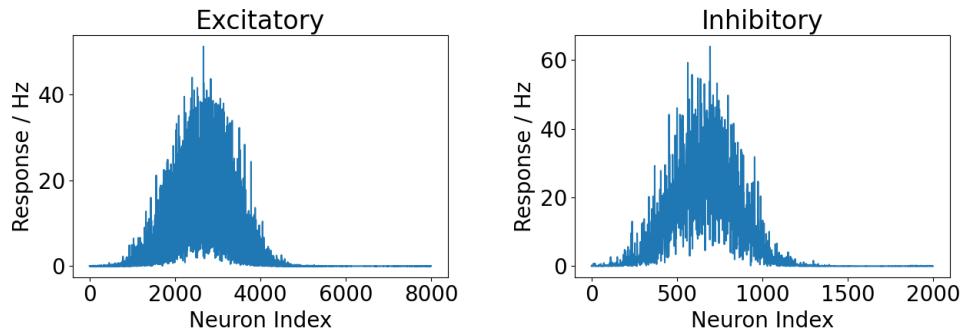


Figure A.10: Neuronal activity from the ground truth parameters used in method validation at an orientation of 60° and contrast of 100%.

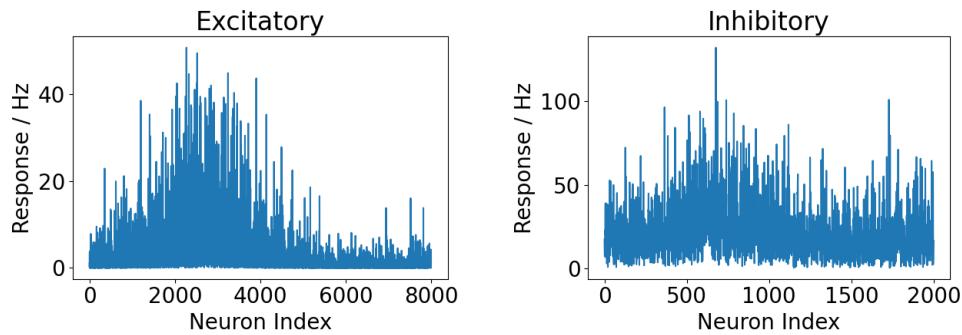


Figure A.11: Neuronal activity from the xNES optimisation using real data at an orientation of 60° and contrast of 100%.

Appendix B

Risk Assessment

Since the project was fully computational, the primary risk was eyestrain from prolonged screen exposure. To mitigate this, regular breaks were taken, and special attention was given to maintaining good posture while working at the computer.