






RESEARCH ARTICLE

Estimation of neural network model parameters from local field potentials (LFPs)

Jan-Eirik W. Skaar¹ , Alexander J. Stasik² , Espen Hagen² , Torbjørn V. Ness¹ , Gaute T. Einevoll^{1,2*} 

1 Faculty of Science and Technology, Norwegian University of Life Sciences, Ås, Norway, **2** Department of Physics, University of Oslo, Oslo, Norway

 These authors contributed equally to this work.

* gaute.einevoll@nmbu.no



Abstract

Most modeling in systems neuroscience has been *descriptive* where neural representations such as ‘receptive fields’, have been found by statistically correlating neural activity to sensory input. In the traditional physics approach to modelling, hypotheses are represented by *mechanistic* models based on the underlying building blocks of the system, and candidate models are validated by comparing with experiments. Until now validation of mechanistic cortical network models has been based on comparison with neuronal spikes, found from the high-frequency part of extracellular electrical potentials. In this computational study we investigated to what extent the low-frequency part of the signal, the local field potential (LFP), can be used to validate and infer properties of mechanistic cortical network models. In particular, we asked the question whether the LFP can be used to accurately estimate synaptic connection weights in the underlying network. We considered the thoroughly analysed Brunel network comprising an excitatory and an inhibitory population of recurrently connected integrate-and-fire (LIF) neurons. This model exhibits a high diversity of spiking network dynamics depending on the values of only three network parameters. The LFP generated by the network was computed using a hybrid scheme where spikes computed from the point-neuron network were replayed on biophysically detailed multicompartmental neurons. We assessed how accurately the three model parameters could be estimated from power spectra of stationary ‘background’ LFP signals by application of convolutional neural nets (CNNs). All network parameters could be very accurately estimated, suggesting that LFPs indeed can be used for network model validation.

OPEN ACCESS

Citation: Skaar J-EW, Stasik AJ, Hagen E, Ness TV, Einevoll GT (2020) Estimation of neural network model parameters from local field potentials (LFPs). *PLoS Comput Biol* 16(3): e1007725. <https://doi.org/10.1371/journal.pcbi.1007725>

Editor: Arnd Roth, University College London, UNITED STATES

Received: March 12, 2019

Accepted: February 12, 2020

Published: March 10, 2020

Copyright: © 2020 Skaar et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the manuscript.

Funding: AJS, EH and GTE received funding from the Research Council of Norway (DigiBrain 248828, CoBra 250128), <https://www.forskningsradet.no/>. TVN and GTE received funding from European Union’s Horizon 2020 Framework Programme for Research and Innovation under Grant Agreements No. 720270 (Human Brain Project SGA1), No. 785907 (Human Brain Project SGA2), <https://ec.europa.eu/programmes/horizon2020/>. The funders had no role in study

Author summary

Most of what we have learned about brain networks *in vivo* has come from the measurement of spikes (action potentials) recorded by extracellular electrodes. The low-frequency part of these signals, the local field potential (LFP), contains unique information about how dendrites in neuronal populations integrate synaptic inputs, but has so far played a lesser role. To investigate whether the LFP can be used to validate network models, we computed LFP signals for a recurrent network model (the Brunel network) for which the

design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

ground-truth parameters are known. By application of convolutional neural nets (CNNs) we found that the synaptic weights indeed could be accurately estimated from 'background' LFP signals, suggesting a future key role for LFP in development of network models.

Introduction

The traditional physics approach to modeling typically involves four steps: (i) A hypothesis is formulated in terms of a candidate *mechanistic* mathematical model, that is, a model based on interactions between building blocks of the system, (ii) predictions of experimentally measurable quantities are calculated from the model, (iii) the predictions are compared with experiments, and (iv) if necessary, the hypothesis is adjusted, that is, a new candidate model is proposed. In neuroscience, a *descriptive* or *statistical* approach has been more common, in particular in systems neuroscience aiming to understand neural network behaviour *in vivo*. Here statistical techniques are used to look, for example, for correlations between measured neural activity and sensory stimuli presented to the animal to estimate receptive fields [1, Ch. 2]. While descriptive models can inform us about neural representations in various brain areas, mechanistic models may explain the biological mechanisms underlying these representations [2].

Starting with Hodgkin and Huxley's development of a mechanistic model for action-potential generation and propagation in squid giant axons [3], mechanistic modeling of neurons is now well established [1, 4, 5]. Numerous biophysically detailed neuron models tailored to model specific neuron types have been constructed, for example, for cells in mammalian sensory cortex [6, 7], hippocampus [8] and thalamus [9, 10]. Further, simplified mechanistic point-neuron models of the integrate-and-fire type excellently mimicking experimental data, have been constructed [11, 12]. At present, mechanistic network models mimicking specific neural circuits are scarce, however. For small networks like the circuit in the crustacean stomatogastric nervous system comprising a few tens of neurons, some excellent models have been developed [13]. For cortical networks important pioneering efforts to construct comprehensive networks with tens of thousands of neurons mimicking cortical columns in mammalian sensory cortices, have been pursued, e.g., [7, 14–17]. These models were found to predict spiking activity in rough qualitative accordance with some observed population phenomena (spiking statistics, spike oscillations, . . .). Fitting of cortical network models to trial-averaged multi-unit activity (MUA) recorded in somatosensory cortex has been pursued for population firing-rate models [18]. However, we do not yet have validated, general-purpose network models that accurately predict experimentally recorded neural activity both in the various 'background' states and as a response to sensory stimulation.

The cortical models above have been compared with experimental spiking activity, that is, the high-frequency part of extracellular electrical potentials. The low-frequency part, the local field potential (LFP), in contrast largely reflects how synaptic inputs are processed by dendrites in the populations of neurons surrounding the electrode contacts [19–21]. Several methods for analysis of cortical LFP signals have been developed, see [20, 21] for reviews. LFP signals have also been used to validate network models both in cortex [22–24] and hippocampus [25–27]. However, a systematic exploration of how informative the LFP signal is for network model validation, has not yet been pursued.

In the present work we explore to what extent the LFP signal generated by a neuronal network model can be used to extract the connectivity parameters of the same network. As a

model network we consider the so-called Brunel network comprising an excitatory and an inhibitory population of recurrently connected integrate-and-fire (LIF) neurons [28]. Point neurons do not generate extracellular potentials, however, and to compute corresponding LFPs we use a hybrid LFP scheme [29]: First the spiking activity is computed by use of the simulator NEST [30], and next the computed spikes are replayed as presynaptic spikes onto biophysically detailed multicompartmental neuron models to compute the LFP using LFPy [31, 32]. The LFP generated by a network depends crucially on the level of temporal correlations of synaptic input onto the neurons [29, 33–35]. Thus the LFPs generated by the Brunel network will, as the spiking activity, vary strongly between the different network states as obtained for different choices of network model parameters.

We assess how well network model parameters can be estimated from the stationary ‘background’ LFP signal. For this, we first train *convolutional neural nets* (CNNs) [36] with LFP training data for which the underlying model parameters are known, and then test the accuracy of parameter estimation on a separate set of LFP test data. As it turns out, a relatively simple CNN is sufficient for the task and is indeed found to accurately estimate the network model parameters. Thus for the present example, the LFP signal contains sufficient information to accurately recover the underlying model parameters. This suggests that not only spiking data, but also LFPs, can be used to validate candidate network models.

Methods

Point-neuron network model

The Brunel network [28] consists of two local populations, one with excitatory and one with inhibitory neurons. These populations of size N_E and N_I , respectively, consist of leaky integrate-and-fire (LIF) neurons interconnected with current-based delta-shaped synapses. Inputs from external connections are modeled as uncorrelated excitatory synaptic input currents with activation governed by a fixed-rate Poisson process with rate v_{ext} .

The sub-threshold dynamics of the point-neurons obey a first-order differential equation, cf. Equation (1) and Equation (2) in Table 1. When the membrane potential of a neuron reaches its firing threshold θ , the neuron emits a spike, the synapses onto all its postsynaptic neurons are activated after a time delay t_d , and the neuron’s membrane potential is clamped to a potential V_{reset} for a refractory period of t_{ref} . Each neuron receives a fixed number of incoming connections (fixed in-degree) from a fraction ϵ of all other local neurons in the network in addition to the external input. The synaptic connection strengths are constant for each population, for excitatory neurons and external input it is given by $J_E = J$ and for inhibitory neurons $J_I = -gJ$. The amount of input the local neurons receive from the external population is determined by the parameter $\eta = v_{\text{ext}}/v_{\text{thr}}$, where $v_{\text{thr}} = \theta/(J\tau_m)$ is the minimum constant rate input that by itself will drive a neuron to its firing threshold, and τ_m is the membrane time constant. A complete description of the point-network model is given in Table 1, with specific parameter values given in Table 2.

Forward-model predictions of LFPs

In order to compute local field potentials (LFPs) from the point-neuron network, we utilized the recently introduced ‘hybrid LFP scheme’ [29] (github.com/INM-6/hybridLFPy), illustrated in Fig 1. The scheme allows for the decoupling of the simulation of spiking dynamics (here computed using point neurons) and predictions of extracellularly recorded LFPs. The latter part relies on reconstructed cell morphologies and multicompartment modeling in combination with an electrostatic forward model. As the complete description of the scheme (including the biophysics-based forward model) and its application with a cortical microcircuit model

Table 1. Description of point-neuron network following the guidelines of [37].

Model summary	
Populations	One excitatory, one inhibitory
Network model	Fixed in-degree, random convergent connections
Neuron model	Local populations: leaky integrate-and-fire, external: Poisson generator
Synapse model	Current-based delta-shaped, fixed strength for each population
Populations	
Names	Excitatory: E Inhibitory: I
Network model	
Connectivity	Fixed number of incoming connections $C_E = \epsilon N_E$ from excitatory population and $C_I = \epsilon N_I$ from inhibitory population
Input	Poissonian synaptic input with fixed rate ν_{ext} for each neuron
Neuron model	
Type	Leaky integrate-and-fire neuron
Description	<p>Dynamics of membrane potential $V_i(t)$ (neuron $i \in [1, N]$):</p> <ul style="list-style-type: none"> • Spike emission at times t_i^j with $V_i(t_i^j) \geq \theta$ • Subthreshold dynamics: $\tau_m \frac{dV_i(t)}{dt} = -V_i(t) + R_m I_i(t) \quad \text{if } \forall l: t \notin (t_i^l, t_i^l + t_{\text{ref}}] \quad (1)$ <p>where τ_m is the membrane time constant, V the membrane potential, R_m the membrane resistance, and I the synaptic inputs.</p> <ul style="list-style-type: none"> • Reset + refractoriness: $V_i(t) = V_{\text{reset}}$ if $\forall l: t \in (t_i^l, t_i^l + t_{\text{ref}}]$ <p>Exact integration with temporal resolution dt</p> <p>Uniform distribution of membrane potentials $V_i \in [V_{\text{reset}}, \theta)$ at $t = 0$</p>
Synapse model	
Type	Delta-shaped postsynaptic current
Description	$R_m I_i(t) = \tau_m \sum_j J_{ij} \sum_l \delta(t - t_i^l - t_d) \quad (2)$ <p>where the first sum is over all the presynaptic neurons j, including the external ones, and the second sum is over the spike times of those neurons. t_i^l is the lth spike of presynaptic neuron j, and t_d is the synaptic delay. δ denotes the Dirac delta function.</p> $J_{ij} = \begin{cases} J, & j \in \{E, E_{\text{ext}}\} \\ -gJ, & j \in \{I\} \end{cases}$ <p>Multapses and autapses are allowed.</p>

<https://doi.org/10.1371/journal.pcbi.1007725.t001>

[15] is given in [29], we here only briefly summarize the main steps taken to predict LFPs from the two-population network described above: To represent each network population we chose one layer-4 pyramidal neuron and one interneuron reconstruction for the excitatory and inhibitory populations, respectively (Fig 1B). The corresponding morphology files L4E_53rpy1_cut.hoc and L4I_oi26rbc1.hoc were also used in [29] (cf. their Table 7), but the apical dendrite of the pyramidal neuron was cut to make it shorter to better fit our smaller column. The somatic positions of all $N_E + N_I$ neurons were drawn randomly with homogeneous probability within a cylinder with radius $r = 564 \mu\text{m}$ and height $\Delta z = 100 \mu\text{m}$, z between -450 and $-350 \mu\text{m}$ (Fig 1B). This depth spread of $100 \mu\text{m}$ is in qualitative agreement with what can be expected for a laminarily organized population in rodent cortices. Each excitatory cell

Table 2. Point-neuron network parameters.

Point-neuron parameters		
Symbol	Description	Value
η	relative amount of external input	[0.8, 4.0]
g	relative strength of inhibitory synapses	[3.5, 8.0]
J	absolute excitatory strength	[0.05, 0.4] mV
τ_m	membrane time constant	20 ms
C_m	membrane capacitance	250 pF
t_d	synaptic delay period	1.5 ms
t_{ref}	absolute refractory period	2 ms
θ	firing threshold	20 mV
V_{reset}	reset membrane potential	10 mV
E_L	passive leak reversal potential	0 mV
N_E	number of excitatory neurons	10000
N_I	number of inhibitory neurons	2500
ϵ	connection probability	0.1
C_E	number of incoming excitatory synapses	1000
C_I	number of incoming inhibitory synapses	250
Simulation parameters		
Training and test data		
T_{sim}	simulation duration	3 s
$T_{transient}$	start-up transient duration	150 ms
dt	time resolution	0.1 ms
Model exploration data		
T_{sim}	simulation duration	30.5 s
$T_{transient}$	start-up transient duration	500 ms
dt	time resolution	0.1 ms

<https://doi.org/10.1371/journal.pcbi.1007725.t002>

morphology was oriented with their apical dendrite pointing upwards in the direction of the positive z -axis and rotated with a random angle around that axis, while inhibitory neurons were rotated randomly around all three axes. The membranes of each morphology were fully passive, with the same membrane time constant τ_m as in the point-neuron network.

In the present hybrid scheme the activity in the LFP-generating populations of multicompartment neurons are obtained by mapping spikes generated by individual LIF neurons in the point-neuron network to synapse activation times at specific positions on their equivalent multicompartment neurons. To obtain the synaptic connectivity onto the different positions on the morphologies of the multicompartment neurons, we defined an ‘upper’ and ‘lower’ layer (homologous to e.g., layer 2/3 and 4) on the depth intervals $[0, z_1]$ and $[z_1, z_2]$, see Fig 1B. The layer-specific connection probability [29, p. 4470–4473] was set so that half of the excitatory synapses onto excitatory neurons were in the upper layer and half in the lower layer. In contrast all inhibitory synapses and excitatory synapses onto inhibitory neurons were positioned in the lower layer. Within each layer, the probabilities for synaptic connections were proportional to the surface area of each compartment normalized by the total compartment surface area within the layer. Only inhibitory synapses were allowed on the soma compartments. The per-neuron synaptic in-degrees were preserved from the network. Since the singular delta-shaped postsynaptic currents (PSCs) cannot be exactly represented in the multicompartment neuron modeling as in the point-neuron network simulations, alpha-function shaped PSCs (Equation (7) in Table 3) with synaptic time constant τ_s were used instead. The

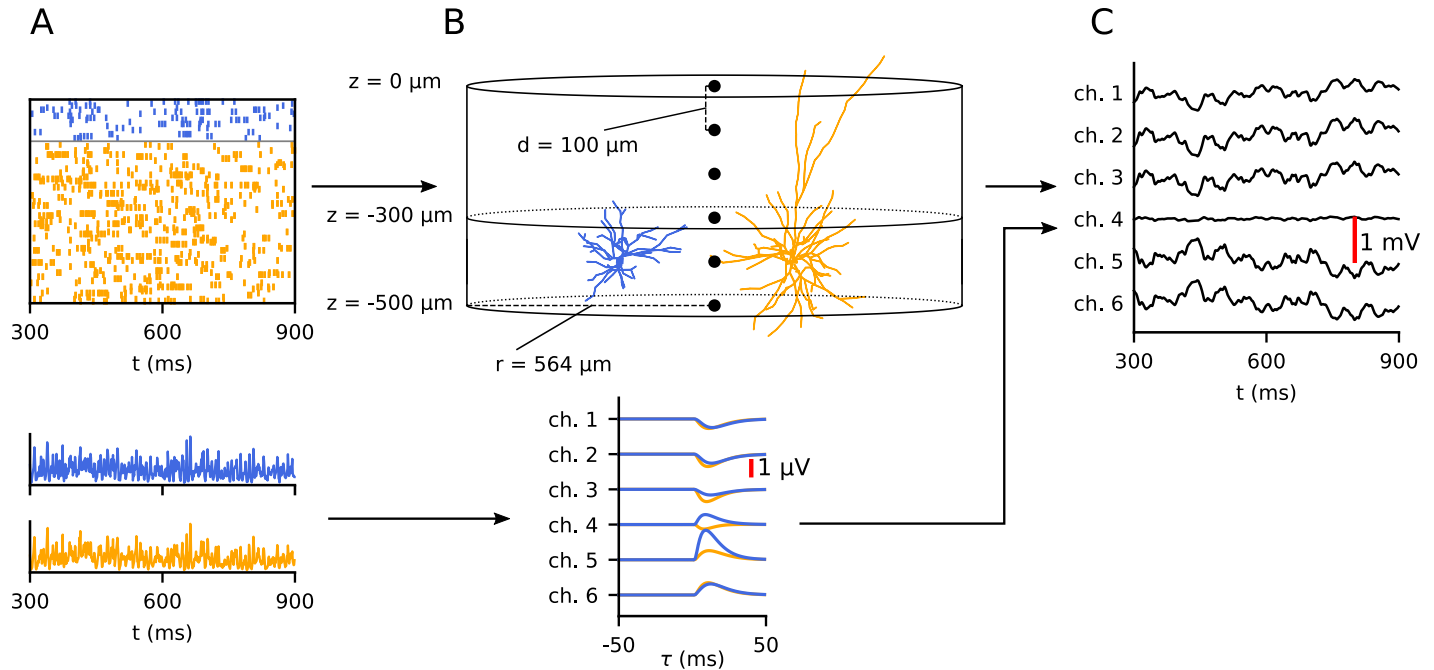


Fig 1. Overview of hybrid scheme for computing local field potentials (LFPs). Top row: First, the dynamics of a network is simulated using a point-neuron simulation (A), and the resulting spike times are saved to file. Orange and blue color indicate excitatory and inhibitory neurons, respectively. In a separate simulation, the obtained spike times are replayed as synaptic input currents onto reconstructed neuron morphologies representing postsynaptic target neurons (B, only one excitatory in orange and one inhibitory neuron in blue are shown). Based on the resulting transmembrane currents of the postsynaptic target neurons in this second simulation, the LFP is calculated (C). Bottom row: Prediction of LFPs from population firing histograms. Instead of running the full hybrid scheme, the LFP can be predicted by the convolution of the population firing histograms (lower figure in A) with kernels representing the average contribution to the LFP by a single spike in each population (lower figure in B). These kernels are computed using the hybrid scheme [29], see Methods.

<https://doi.org/10.1371/journal.pcbi.1007725.g001>

amplitude of the PSCs was chosen so that the total transferred charge is equal for both synapse types (thus preserving the total synaptic input current between the network and multicompartment neurons). A full description of the multi-compartment neuron model is given in Tables 3 and 4.

The presently used choice of current-based synapses and morphologies with passive membranes in the multicompartment neuron models introduces a linear relationship between any presynaptic spike event and contributions to the LFP resulting from evoked currents in all postsynaptic multicompartment neurons. Thus the LFP contribution $\phi_Y^j(\mathbf{r}, t)$ at position \mathbf{r} from a single presynaptic point-neuron neuron j in population Y can, in general, be calculated by the convolution of its spike train $v_Y^j(t) \equiv \sum_k \delta(t - t_j^k)$ with a unique kernel $H_{YX}^j(\mathbf{r}, \tau)$ as $\phi_Y^j(\mathbf{r}, t) = \sum_X (v_Y^j * H_{YX}^j)(\mathbf{r}, t)$. This kernel encompasses effects of the postsynaptic neuron morphologies and biophysics, the electrostatic forward model, the synaptic connectivity pattern, conduction delay and PSCs. (Note that to be in accordance with the notation used in [28], we here use a different notation than in [29] where instead i and X denoted presynaptic neurons, and j and Y denoted postsynaptic neurons.)

The resulting LFP due to spikes in a presynaptic population Y is then given by [29]

$$\phi_Y(\mathbf{r}, t) = \sum_X \sum_{j \in Y} (v_Y^j * H_{YX}^j)(\mathbf{r}, t) . \tag{9}$$

The evaluation of this sum is computationally expensive for large population sizes. For our

Table 3. Description of multi-compartment neuron populations.

Model summary	
Populations	Local excitatory and inhibitory populations
Neuron model	Multi-compartment neurons with passive cable formalism
Synapse model	Current-based, α -function shaped, fixed strength for each population
Topology	Cylinder of 1 mm^2 cross-section with somas of both populations positioned in single layer of thickness 0.1 mm .
Neuron models	
Type	Reconstructed multi-compartment morphologies with passive electrical properties
Description	For each neuron, the membrane potential V_n of compartment n connected to m other compartments k , with a surface area a_n , length l_n and diameter d_n is given by: <div style="text-align: center;"> $\sum_{k=1}^m g_{akn}(V_k - V_n) = C_{mn} \frac{dV_n}{dt} + I_{mn} \tag{3}$ $C_{mn} = c_m a_n \tag{4}$ $g_{akn} = \pi(d_n^2 + d_k^2)/(4r_a(l_n + l_k)) \tag{5}$ $I_{mn} = g_{Ln}(V_n - E_L) + \sum_j I_{jn} , \tag{6}$ </div> <p>where for compartment n, C_{mn} is the membrane capacitance, g_{akn} the axial conductance from compartment k, I_{mn} the membrane current, g_{Ln} the membrane leak conductance, E_L the extracellular reversal potential, and I_{jn} the synaptic current from presynaptic neuron j.</p>
Synapse model	
Synapse type	α -function shaped postsynaptic current
Description	$I(t) = H(t - t_a)JCe^{-(t-t_a)/\tau_s} \tag{7}$ $H(t) = 0 \text{ for } t \leq 0, \text{ otherwise } 1. \tag{8}$ <p>Here t_a is the activation time of the synapse, J the synaptic strength, and τ_s is the synaptic time constant. C is a constant chosen so that $JC \int_0^\infty te^{-(t-t_a)/\tau_s} dt = C_m J$, assuring that the same total charge is transferred as in the δ-function synapse in the point-neuron network.</p>
Topology	
Type	Cylinder with radius $1/\sqrt{\pi}$ mm and height 0.5 mm containing two vertical sections
Description	<ul style="list-style-type: none"> • Cylinder extends from $z = -500 \mu\text{m}$ to $z = 0$ • All somas are randomly placed with a uniform distribution within the boundaries $r \leq 564 \mu\text{m}$ and $-450 \mu\text{m} \leq z \leq -350 \mu\text{m}$ • Two regions separated by the plane $z = -300 \mu\text{m}$ • Synapses on inhibitory neurons are placed in lower region • Inhibitory synapses on excitatory neurons are placed in lower region • Excitatory synapses on excitatory neurons are split equally between regions

<https://doi.org/10.1371/journal.pcbi.1007725.t003>

purposes where the calculation of LFP signals lasting seconds must be repeated tens of thousands of times to have training and test data for the CNNs, this scheme is not feasible.

Following [29, Fig 13] we instead use a firing-rate approximation and compute the LFP by a convolution of population firing rates $v_Y(t) \equiv \sum_{j \in Y} v_Y^j(t)$ and averaged kernels

$\bar{H}_Y(\mathbf{r}, \tau) \equiv 1/N_Y \sum_X \sum_{j \in Y} H_{YX}^j(\mathbf{r}, \tau)$, that is,

$$\phi_Y(\mathbf{r}, t) = (v_Y * \bar{H}_Y)(\mathbf{r}, t) . \tag{10}$$

As in [29], these averaged kernels $\bar{H}_Y(\mathbf{r}, \tau)$ were computed using the full hybrid-scheme. This was done by computing the LFP resulting from a fully synchronous activation of all the outgoing synapses from all neurons in the presynaptic population. Thus for the computation of the

Table 4. Multi-compartment neuron parameters.

Multi-compartment neuron parameters		
Symbol	Description	Value
τ_m	membrane time constant	20 ms
C_m	membrane capacitance	1.0 $\mu\text{F}/\text{cm}^2$
R_m	membrane resistivity	τ_m/C_m
R_a	axial resistivity	150 Ωcm
τ_s	synaptic time constant	5 ms
E_L	passive leak reversal potential	0 mV
V_{init}	membrane potentials at $t = 0$ ms	0 mV
σ_e	extracellular conductivity	0.3 Sm^{-1}

<https://doi.org/10.1371/journal.pcbi.1007725.t004>

LFP kernel, we have $v_Y^i(t) \equiv \delta(t - t_Y)$ where t_Y is the timing of the synchronous event in population Y . In the application of Eq 10, this computed kernel is then convolved with the population firing rates measured in the point-neuron simulations to compute the LFP.

By using this kernel approach the computational resources needed to run LFP simulations are reduced by several orders of magnitude compared to direct use of Eq 9. To test the accuracy of the approximation of using Eq 10 instead of Eq 9, we compared their LFP predictions for a set of example parameter sets and found in general excellent agreement between the resulting power spectra. A comparison is shown in the lower panels of the first figure in Results.

The kernel \bar{H}_Y will scale linearly with the postsynaptic strengths of population Y , and is therefore dependent on the parameters J for $Y \in \{E, I\}$ and g for $Y \in \{I\}$. The kernels were thus computed only once for a set of reference values for J and g , and for each simulation these reference kernels were scaled accordingly to the particular values of J and g . The LFP was computed across depth through the center of the cylindrical volume with a spatial resolution d as illustrated in Fig 1B for the same duration as the network simulations.

Statistical methods

LFP spectral analysis. The power spectral densities $P_\phi(\mathbf{r}, f)$ of LFPs $\phi(\mathbf{r}, t)$ in each location \mathbf{r} were estimated using Welch's average periodogram method [38]. For this we used the implementation from the Python SciPy package [39] (`scipy.signal.welch`), with parameters listed in Table 5.

Statistical measures of activity. Two statistical measures were employed to probe the spiking network activity in the different regions of the parameter space. Simulations of 30.5 seconds of the activity were run and used to calculate the statistics, where the first 500 ms of the simulations were discarded. These simulations were run in addition to the ones entering the training and testing of the convolutional neural network (described in the next section), in

Table 5. Parameters for Welch's method for computing power spectral density (PSD) of LFP.

Power spectrum estimation		
Symbol	Description	Value
$NFFT$	number of data points used in each window	300
$noverlap$	number of overlapping data points	150
F_s	sampling frequency	1 kHz
$window$	window function	Hann

<https://doi.org/10.1371/journal.pcbi.1007725.t005>

order to have longer data sets for calculating reliable statistics characterizing the parameter regimes.

The mean network firing rate, including both the excitatory and inhibitory populations was calculated as

$$\bar{v} = \frac{1}{(N_E + N_I)(T_{\text{sim}} - T_{\text{transient}})} \sum_i \sum_l \int_{T_{\text{transient}}}^{T_{\text{sim}}} \delta(t - t_l^i) dt, \quad (11)$$

over all neurons i and their spikes l at spike times t_l^i . The coefficient of variation (CV) of the inter-spike intervals (ISI) of individual neurons was used as a measure of the irregularity of firing [40]. The presently used mean CV was defined as

$$\overline{CV} = \frac{1}{N} \sum_i \frac{\sigma_{\text{ISI},i}}{\mu_{\text{ISI},i}}, \quad (12)$$

averaged over all neurons i .

As a measure of the degree to which the LFP power spectrum is spread out over different frequencies, we employed the entropy of the normalized power spectrum of the LFP measured in the uppermost channel, defined as

$$S = - \sum_n \tilde{P}_\phi(f_n) \log \tilde{P}_\phi(f_n), \quad (13)$$

where $\tilde{P}_\phi(f_n)$ is the power spectrum of the LFP $\phi(\mathbf{r}, t)$ at frequency f_n normalized to unity.

Since the power spectrum is computed numerically using Welch's method, this introduces a discretization in frequency space.

Simulation of training and test data

Two different sets of training and test data were created for this study. The first data set is from a large parameter space with $\eta \in [0.8, 4.0]$, $g \in [3.5, 8.0]$ and $J \in [0.05, 0.4]$ mV. The parameters were drawn randomly from a uniform probability distribution with a total of 50000 examples. This data set encompasses all four different activity states displayed by the Brunel network, illustrated in Fig 2. These activity states include synchronous regular (SR), asynchronous irregular (AI), synchronous irregular (SI) with either slow or fast oscillations. The large parameter space is illustrated by the orange outline in Fig 2. The second data set is from a smaller parameter space with $\eta \in [1.5, 3.0]$, $g \in [4.5, 6.0]$ and $J \in [0.1, 0.25]$ mV, encompassing only the AI activity state, as illustrated by the blue outline in Fig 2. All other parameters (Table 2) were kept constant across simulations, which were run for a duration of $T_{\text{sim}} = 3$ s. Start-up transients with duration $T_{\text{transient}} = 150$ ms were discarded. LFP signals for all spiking output were computed as outlined above, and as final training and test data we estimated the power spectrum $P_\phi(\mathbf{r}, f)$ in each LFP channel.

Parameter estimation by convolutional neural networks

The CNN architecture is illustrated in Fig 3 and fully described in Table 6, and was set up using the Keras machine learning framework [41] running on top of TensorFlow [42]. It consisted of three convolutional layers with 20 filters, each followed by max pooling layers, and two fully connected layers before the output layer. The rectified linear unit (ReLU) function $f(x) = \max(0, x)$ was used as the activation function for all layers apart from the output layer, and biases were only used in the fully connected layers. As input, it took the PSD of each LFP channel, a 6 by 151 matrix. The convolutions were done in one dimension, with kernels extending over all LFP channels. There were two fully connected layers, with 128 nodes each,

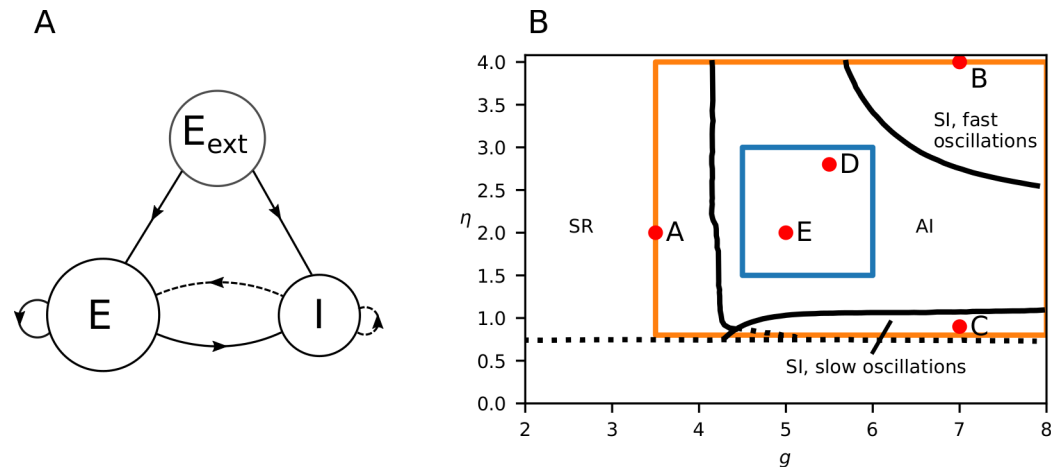


Fig 2. Brunel model network and phase diagram. **A**, Illustration of network. Solid lines represent excitatory connections, dashed lines inhibitory connections. **B**, Phase diagram, adapted from [28], Fig 2A. Different network states arise depending on the parameters $\eta = v_{ext}/v_{thr}$ and g (where in the present example a fixed synaptic delay t_d of 1.5 ms is used). SR stands for synchronous regular, SI for synchronous irregular, and AI asynchronous irregular. Orange box shows the extent of parameters we simulated and blue box when we restricted the simulations to the AI state. Note that this plot shows a slice of the parameter space for a given value of $J = 0.1$. We considered different values of J in the study, so the actual parameter space is a cube, with the third axis being in the J -direction. The red dots labeled A–E indicate the η and g values of the example activities shown in the first figure in Results.

<https://doi.org/10.1371/journal.pcbi.1007725.g002>

before the output layer consisting of 3 nodes. Each node in the output layer corresponded to a single parameter η , g and J .

The LFP PSD was normalized for each channel by the mean of the sum of the PSD over all frequencies, serving to diminish the variation in amplitude across the different LFP PSDs input to the network, while keeping the variation in amplitude across channels for each single LFP PSD. Each parameter range was linearly mapped to the interval [0, 1] when used as target

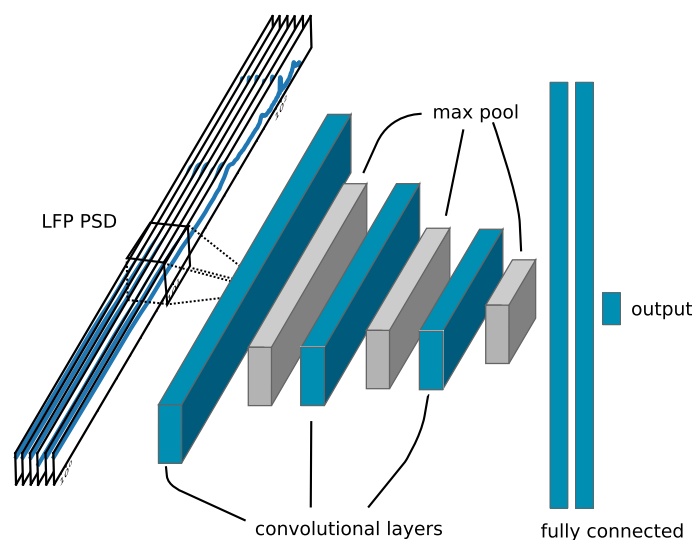


Fig 3. Illustration of convolutional neural network (CNN). The PSDs of all six LFP channels are taken as input. The three convolutional layers consist of 20 filters each, and are followed by max pooling. Two fully connected layers precede the output layer which consists of 3 nodes, one for each parameter.

<https://doi.org/10.1371/journal.pcbi.1007725.g003>

Table 6. Detailed specification of presently used convolutional neural network (CNN). The convolutional kernel dimensions are given as [frequency, channels in, channels out], the strides and window sizes are given in the frequency dimension.

Convolutional neural network	
Layer	Description
Conv. layer 1	kernel size: 12x6x20
	stride: 1
	activation: ReLU
	bias: no
Max pool 1	initialization: glorot uniform
	window size: 2
Conv. layer 2	stride: 2
	kernel size: 3x20x20
	stride: 1
	activation: ReLU
Max pool 2	bias: no
	initialization: glorot uniform
Conv. layer 3	window size: 2
	stride: 2
Max pool 3	kernel size: 3x20x20
	stride: 1
	activation: ReLU
	bias: no
Dense layer 1	initialization: glorot uniform
	window size: 2
Dense layer 2	stride: 2
	nodes: 128
	activation: ReLU
	bias: yes
Output layer	initialization: glorot uniform
	nodes: 128
	activation: ReLU
Output layer	bias: yes
	initialization: glorot uniform
	nodes: 3
	activation: Linear
Output layer	bias: no
	initialization: glorot uniform
	initialization: glorot uniform

<https://doi.org/10.1371/journal.pcbi.1007725.t006>

quantities during training and inference. The network was trained by batch gradient descent on 40000 of the simulated LFPs, while the final 10000 simulated LFPs were reserved for testing. To train the CNN, we required a *loss function* which was minimized during training. We defined the *loss* as the mean squared error of the estimator

$$loss = \langle (\hat{a} - a_{\text{true}})^2 \rangle \quad (14)$$

where \hat{a} is the estimate (output from the CNN) and a_{true} is the truth ('ground-truth' value) of any vector of normalized network parameters a .

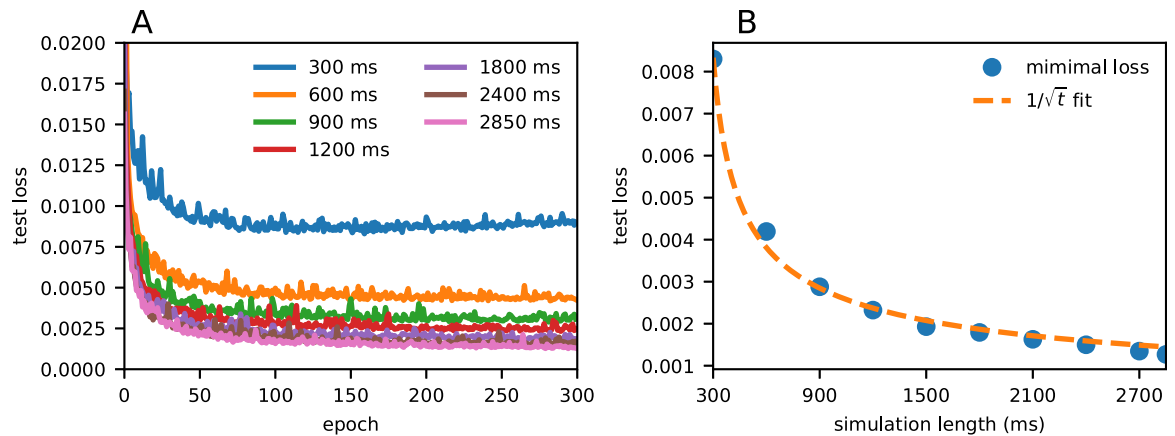


Fig 4. A, Test loss as a function of number of training epochs of the CNN for different simulation lengths. B, Minimal loss (that is, smallest loss in panel A) as a function of simulation length. A function with $\sim 1/\sqrt{t}$ shape was fitted to the data to illustrate that the scaling is explained by limited amount of data and that the error decreases as more data is used. The R^2 score was 0.994.

<https://doi.org/10.1371/journal.pcbi.1007725.g004>

The Adam optimizer [43] was used, with a batch size of 100, learning rate of 0.001 and β_1 , β_2 and ϵ_{Adam} parameters of 0.9, 0.999 and 10^{-8} respectively. The networks were trained for 400 epochs, and the network weights with the lowest test loss were saved.

Effect of duration of LFP signals

It was *a priori* not known what duration of the data are required to obtain stable results. To test this, the duration of each LFP simulation was successively extended, the PSD of the LFP was computed using the Welch method, and the CNN was trained with the data to predict the three parameters simultaneously. The test loss during training is shown in Fig 4A.

Overall, the loss decreased with training duration and reached a plateau after a certain amount of training epochs. Increase of the duration of each simulation led to a decrease in the loss. This is due to lower noise in the estimated PSDs when it is computed over larger time windows.

The results in the figure suggested that a simulation duration of about 1800 ms would be a good choice, as shorter simulation times decreased the performance. Fig 4B shows the scaling of the minimal test loss (that is, loss obtained in the limit where more training epochs do not improve results) as a function of simulation duration. The $\sim 1/\sqrt{t}$ least squares fit was motivated by the scaling of the error of the mean, which gives the square root dependence of the standard error of the mean. This scaling assumes uncorrelated experiments, which is not the case when using Welch's method as we do. Nevertheless, the fact that this scaling was observed to hold in our estimations suggested that the error indeed is largely explained by limited amounts of data and that the error decreases as more data is used.

Technical details

Reproducibility. The simulations were done using Python v2.7.12. All point-network simulations were done with the NEST simulator v2.12.0 [30]. The forward-modeling of the LFP was done using hybridLFPy v0.1.3 [29], with NEURON v7.5 [44]. All simulations were run on the Stallo high-performance computing cluster consisting of 2.6 GHz Intel Xeon E5 2670 and 2.8 GHz Intel Xeon E5 2680 CPUs.

The convolutional neural networks were trained using Python v3.5.2 using Keras v2.2 with TensorFlow v1.10.0 as backend. All simulation and analysis code for this study is available on Github: https://github.com/CINPLA/Skaar_et_al_2020_PLoS_Comput_Biol.

Results

The aim of this study is to investigate the possibility of estimating network model parameters for the Brunel two-population spiking-network model [28] from the stationary ‘background’ LFP signal. We start by describing this spiking model and its salient dynamical properties and further describe how the resulting spikes can be used in a hybrid scheme to calculate associated LFPs [29]. Then we discuss the estimation performance of a convolutional neural network (CNN) to predict network parameters of the Brunel network model based on LFP data only.

Network model and LFPs

The presently used Brunel network produces four different network states dependent on the post-synaptic potential amplitude of excitatory connections J , the ratio of inhibitory to excitatory connection strength g , and the strength of the external input η relative to the threshold rate, see Fig 2. In the synchronous regular (SR) state the neurons fire regularly and in synchrony. In the asynchronous irregular (AI) state individual neurons have irregular firing and very little synchronization. The synchronous irregular (SI) state, characterised by oscillatory population firing rates, displays highly irregular firing of individual neurons. Example spike raster plots and population firing rates for SR, AI and SI states of the network are shown in the top rows of Fig 5. AI states are commonly believed to be realised in most healthy neural networks *in vivo*, often characterized by low average pairwise spike-train correlations (see e.g., [45]) and irregular spike trains (see e.g., [46]).

To generate training and test data for the convolutional neural net (CNN), we simulated the network for many combinations of parameters ($\eta \in [0.8, 4]$, $g \in [3.5, 8]$ and $J \in [0.05, 0.4]$ mV). This parameter space includes parameter combinations giving three of the states described above: AI, SI, and SR (see orange rectangle in Fig 2). For details of the simulation procedure, see Methods.

The LFPs were simulated using the so-called hybrid scheme introduced by [29]. In this scheme, neuronal network activity is predicted by a point-neuron network (here the Brunel network), and the corresponding LFPs are estimated in a subsequent step by ‘playing back’ spike times as activation times of synapses distributed across reconstructed neuron morphologies representative for each population type. The LFP is then computed from the resulting transmembrane currents combined with an electrostatic forward model derived from volume-conductor theory, as detailed in Methods. An overview over the hybrid scheme, including the geometrical organisation of the ‘cortical column’ used in the LFP-generating step, is shown in Fig 1.

Example LFPs for different network states. In the presently used model set-up, the LFP is linearly dependent on the point-neuron network spiking activity. Any network parameter change affecting ongoing spiking activity will therefore directly affect the LFP. The panels in the lower two rows of Fig 5 show the resulting LFP and corresponding LFP power spectra for five different network parameter combinations of η , g and J .

An example synchronous regular state (SR) is shown in panel A. The simulation showed high regularity and synchrony of the individual spike trains and a strongly oscillating population firing rate. The corresponding LFP generally had a similar time course over all channels, though with opposite phases for the topmost and lower recording channels. The power spectral density (PSD) of the LFP showed a decrease in power with increasing frequency, though

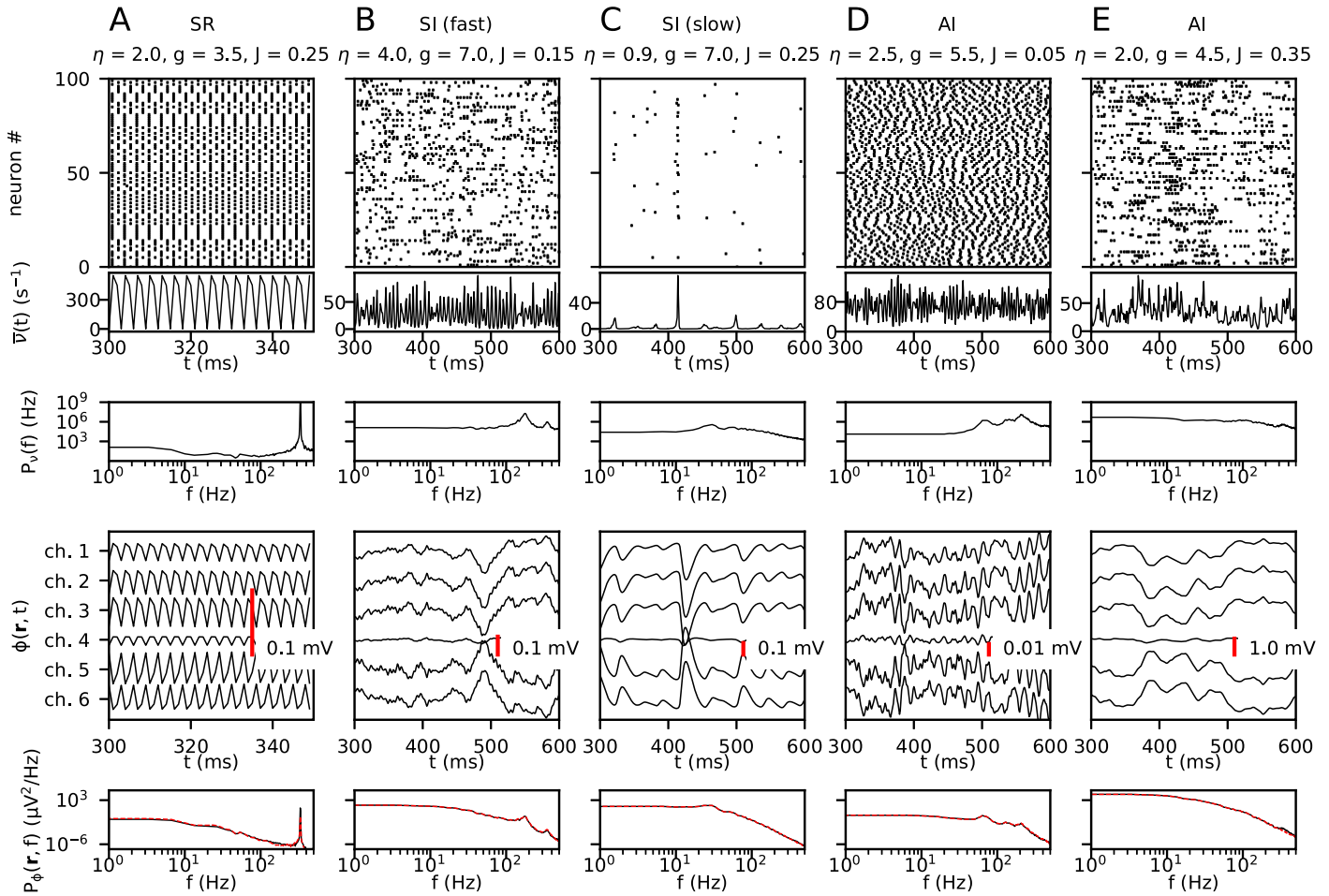


Fig 5. Examples of simulated spiking network activity and LFPs for different sets of network parameters (η , g and J). For each simulation, A–E, the first row shows spike trains from 100 randomly selected neurons across both populations. The second and third row show the population firing rate (including both the excitatory and inhibitory neurons) and its power spectral density (PSD). The final two rows show the LFP signal from all six channels and the PSD of channel 1, respectively. The dashed red lines in the lowest panel shows the LFP PSD computed from spikes in individual neurons (Eq 9) rather than with the presently used population firing-rate approach (Eq 10, black lines) which is computationally much less demanding. In general, the agreement is seen to be very high, the only discrepancy is seen for the SR-state example where the height of the peak around 300 Hz differs. The network states for the five examples (SR/SI/fast)/SI(slow)/AI, see text) are indicated at the top.

<https://doi.org/10.1371/journal.pcbi.1007725.g005>

with a clear peak at around 330 Hz. This peak was also seen in the PSD of the firing-rate, reflecting the tight relationship between spikes and LFPs.

Two examples of the synchronous irregular state (SI) are illustrated in Fig 5B and 5C, characterised by synchrony of the firing of neurons while individual neurons fire irregularly. In panel B an example with high firing and fast oscillations is shown. Here the power spectrum of the LFP showed two peaks at around 175 and 350 Hz, respectively. Again, the same peaks are found also in the firing rate spectra. In contrast, panel C shows a low-firing SI state with more slowly varying population firing rates, and with a small peak around 30 Hz in the firing-rate and LFP power spectra.

Two examples of the asynchronous irregular state (AI) are shown in the last two panels (Fig 5D and 5E). As suggested by the name, this state is defined by lack of synchrony between different neurons and irregular firing patterns of each neuron. For the example in panel D, the firing-rate PSD exhibited three high-frequency peaks, with the peak at the highest frequency (~ 200 Hz) being highest. The same three peaks were found also in the LFP PSD, but now the

peak at the lowest frequency (~ 70 Hz) was highest. This reflects low-pass filtering effects of the LFP from synaptic and intrinsic dendritic filtering [34, 47]. In panel E the recurrent excitation J is much increased compared to the example in panel D. This combined with a reduction of the relative inhibition g , gave much larger LFP signals, as reflected in the high power of the LFP seen for the low frequencies in the LFP PSD. For this parameter set neither the firing-rate PSD nor the LFP PSD exhibited any notable peaks.

Model behaviour across parameter space. To extract network model parameters from recordings of neural activity such as the LFP, the network model parameters must necessarily be reflected in these recordings. After the qualitative discussion above, we proceed to discuss how the network behaves over the entire parameter space. We therefore give an overview of how different spike- and LFP-based measures of neural activity vary across parameter space.

Spikes.

Panel A in Fig 6 shows how the mean network firing rate varied over the parameter space. The overall trend was that with increasing g , the firing rate decreased since inhibition was

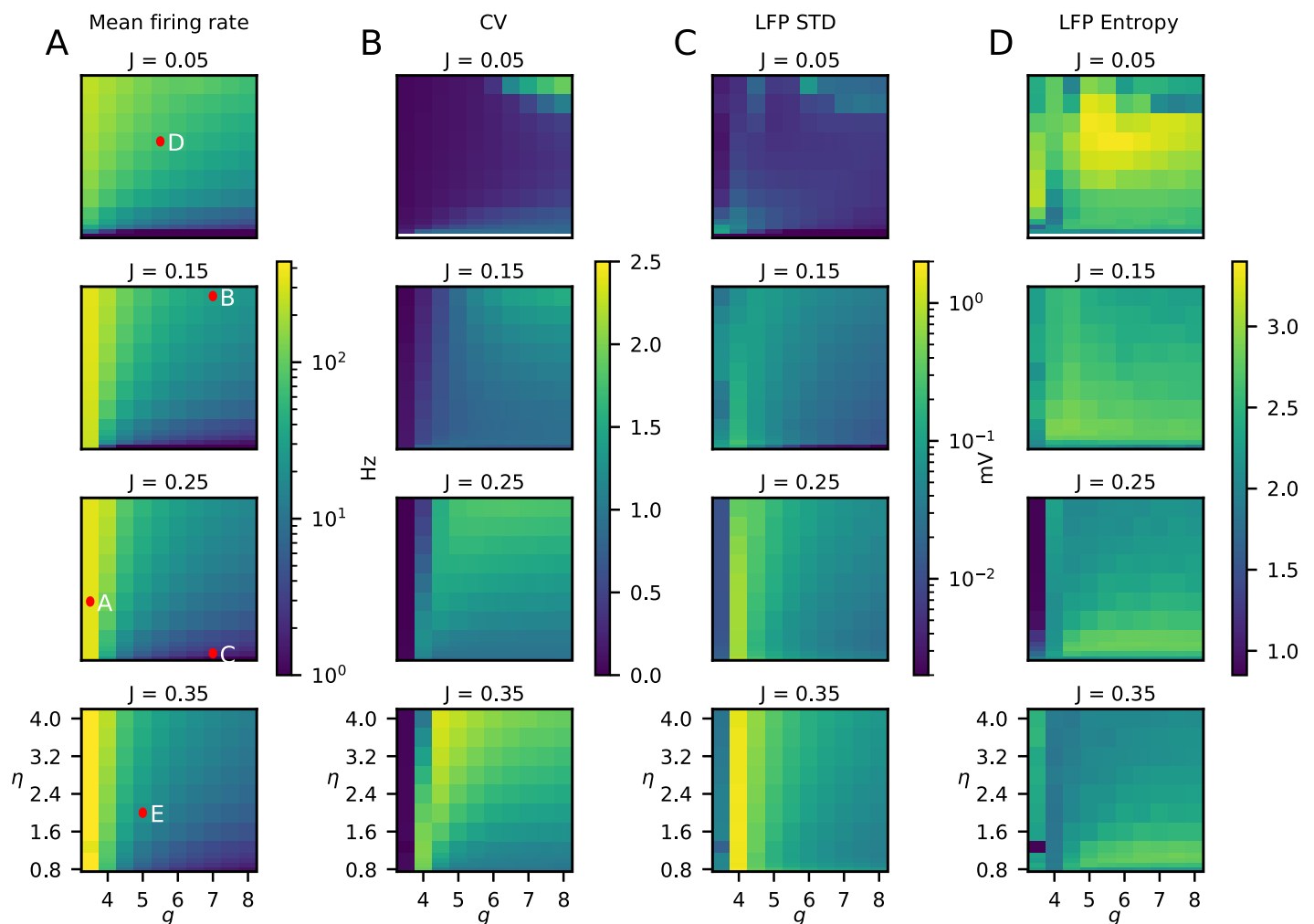


Fig 6. Statistical measures of network activity for different combinations of network parameters (η , g and J). A, Average population firing rates, that is, average firing rate over all neurons and times. The red dots show the parameter values of the examples in Fig 5. B, Mean coefficient of variation (Eq (12)) of the inter-spike intervals over all neurons as a measure of the spiking irregularity. C, Square root of the variance of the LFP signal integrated over time for the topmost channel (channel 1). This measure corresponds to the square root of integral of the power spectrum of the LFP over all frequencies [33], and is referred to as the standard deviation of the LFP (LFP STD). D, LFP Entropy, cf. Eq 13.

<https://doi.org/10.1371/journal.pcbi.1007725.g006>

increased. The transition at $g = 4$ resulted from the fact that there are four times more excitatory neurons than inhibitory, and thus for $g < 4$ excitation dominates network behaviour. For $J \geq 0.15$, three separable regions with smooth transitions emerged: A region of high firing rate on the left border of the plots ($g \lesssim 4$), a region of low firing rate on the bottom of the plots ($g \geq 5, \eta \lesssim 1$), and a region of intermediate firing rate in the upper right of the parameter space. For smaller values of J ($J \lesssim 0.1$) the transition between the high-firing region and the intermediate-firing region became smoother. Thus, large values of J amplified the differences between the regions. These distinct regions in the firing-rate phase diagram correspond well with the phase diagrams derived by [28], see Fig 2.

Panel B in Fig 6 correspondingly displays the parameter dependence of the average coefficient of variation (CV) of the inter-spike intervals. Similar to the population firing rate, one can see a boundary at about $g \approx 4$ over a large part of the considered parameter range of J and η . In the region with low η and high g ($\eta \lesssim 1.5, g \geq 5$) there was also a distinct area with low CV, reflecting the expected lower CV of the slow-oscillation SI state compared to the AI state. For small values of J ($J \lesssim 0.1$), there was a region of larger CV visible in the upper right corner of the parameter space ($g \geq 6$ and $\eta \geq 3.6$). This region overlaps with fast-oscillation SI state described by [28] (see phase diagram in Fig 2).

LFP.

The example LFP patterns in Fig 5 showed substantial variability of the LFPs for different network parameter values. This suggests that it indeed may be possible to estimate network parameters from the LFP. To explore this in more detail, we show in panels C and D of Fig 6 two different measures of LFP signals across the same parameter space.

Panel C shows a measure of the overall signal power of the LFP signal, that is, the standard deviation (STD) for the topmost channel (channel 1). This measure corresponds to the square root of the variance of the LFP signal integrated over all frequencies [33]. In panel C a first observation was that large values of the excitatory weight J led to higher values of the LFP STD, not surprising given the stronger excitatory synaptic inputs. Likewise, it was seen that the LFP STD generally decreased when inhibition, that is, g , increased. Interestingly, despite the very high firing activity for values of g smaller than 4, the LFP STD was small for these parameter values. This can be understood by inspection of panel A in Fig 5 which shows results for an example state with $g = 3.5$: Even if there are strong bombardments of synaptic inputs onto the LFP-generating excitatory neurons, the input is so clock-like and regular that there is little power in the LFP signal at the lower frequencies. The only strong LFP signal contribution was obtained for frequencies over ~ 300 Hz, corresponding to the peak seen in the firing rate PSD.

The LFP STD in panel C measures the square root of the squared LFP signal across time or equivalently, according to Parseval's theorem, essentially the square root of the integral of the LFP PSD across frequency. In contrast the measure labeled 'LFP Entropy' in panel D measures how much the overall LFP power is spread across the different frequencies, cf. Eq 13 in Methods. The largest entropy value was observed for the smallest excitatory weight ($J = 0.05$ mV), but the detailed parameter dependence of the LFP entropy was not the main point here. The most important observation was that the parameter dependence of LFP Entropy was qualitatively different from the parameter dependence of LFP STD. This illustrates that the frequency-resolved PSD likely contains more information regarding the underlying network parameters than either the overall amplitude (LFP STD) or the frequency spread (LFP Entropy) alone.

Network parameters are accurately estimated from LFP

After this rough survey over how the LFP for the Brunel network model varies across parameter space, we now ask the question: Can the network parameters be estimated from this LFP by

use of Convolutional Neural Networks (CNNs)? We chose to use CNNs because they do not rely on manual feature extraction, and our analysis thus does not depend on any assumption of how the model network parameters are reflected in the LFP. Further, we used the power spectral density (PSD) of the LFP for this analysis, that is, used the PSD as input to the CNNs. This approach removes phase information in the LFP. However, since we only considered LFP data from stationary network activity, the hypothesis was that most of the available relevant information regarding network parameters should be contained in the PSD.

Our CNN consisted of three convolutional layers followed by two fully connected layers. An illustration can be seen in Fig 3, and detailed specifications are given in Methods. We generated several pairs of training and testing data sets for different scenarios. The parameter space was both sampled randomly and on a regular grid. We also generated a training and test data set on a subset of the parameter space, but with the equal amount of simulations.

While several approaches were tested and compared, we defined the following set-up as the standard set-up: The data was simulated using randomly distributed parameters η , g and J with a simulated duration of 2.85 seconds for each trial. From the simulated LFP, the power spectral densities (PSD) for six recording channels were computed and used as input to the CNN. Then, a single CNN network was trained to predict the parameter vector $\vec{p} = (\eta, g, J)$ simultaneously, and all three parameters were set to contribute equally to the loss function, Eq 14. To achieve this, the parameter ranges of η , g and J were all scaled to the unit interval $[0, 1]$ for the considered part of the parameter space.

To quantify and illustrate the accuracy of the parameter estimation we used the estimation error $\hat{a} - a_{\text{true}}$ where a_{true} was the true value and \hat{a} the estimated value. Fig 7 (orange lines) shows the accuracy of the three network parameters when considering the full parameter space ($\eta \in [0.8, 4)$, $g \in [3.5, 8)$ and $J \in [0.05, 0.4)$ mV). As observed, the estimation errors are in all cases generally smaller than 5%. Further, the estimates had small biases, that is, the mean errors were close to zero (see also Table 7).

The full parameter space considered above covered four of the characteristic network states seen for the Brunel network, see orange rectangle in Fig 2. Here the network-generated LFP can be expected to vary substantially across parameter space making the CNN estimation easier. We thus next explored to what extent CNNs could estimate network parameters within a particular state, that is, the AI state which is thought to be most relevant for cortex.

Training and validation of the CNN were repeated using a second data set, fully contained within the AI region ($\eta \in [1.5, 3)$, $g \in [4.5, 6)$ and $J \in [0.1, 0.25)$ mV), see blue rectangle in Fig 2. The same amount of training and test data were used as for the full parameter space, so effectively the restricted parameter space was more densely sampled. Estimation errors are shown in Fig 7 (blue lines). With a same-sized data set containing only the AI state, the observed error was even smaller than for the full parameter space. Thus focusing on a single network state within which there expectedly is less variation in the LFP, increased the accuracy. However, when using the CNN trained with the data from the *full* parameter space, the estimation accuracy for a restricted test set containing only the AI state, was reduced (Fig 7, purple lines). The accuracy was still better than when estimating parameters across the full parameter space, though, that is, the purple line was always positioned between the yellow and blue lines in the cumulative plots in Fig 7B. Further, independent of which data set was used, the g parameter was always the one with the largest prediction accuracy compared with η and J .

Highest prediction accuracy of network parameters in AI state

Next, the variation of the parameter estimation errors across the full parameter set was investigated (Fig 8A–8C). The estimation of η (panel A of Fig 8) was less reliable in the region of low

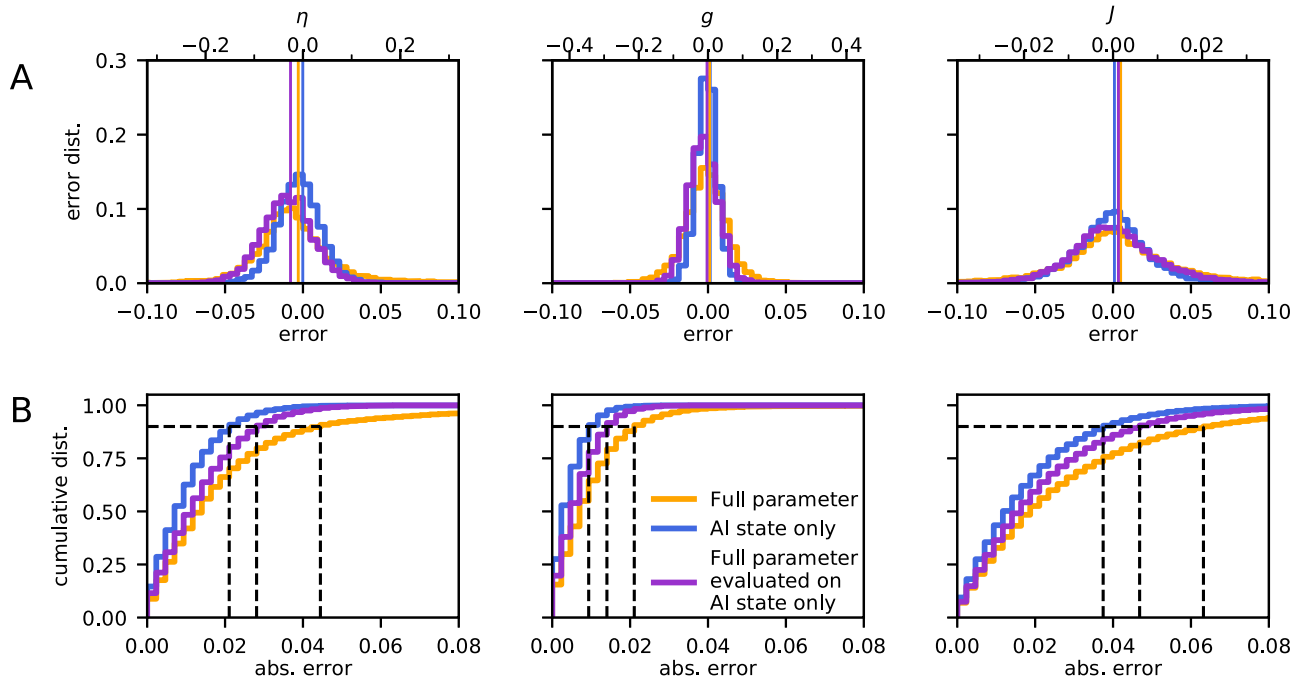


Fig 7. Accuracy of network parameter estimation. A, Estimation error distributions for η , g and J averaged over the entire parameter space. In the plots all parameter ranges were rescaled to the interval $[0, 1]$ for easier comparison on the lower x -axis, the upper x -axis shows the original values. The vertical line indicates the mean of both distributions. The orange curve shows the result when using the full parameter set ($\eta \in [0.8, 4]$, $g \in [3.5, 8]$ and $J \in [0.05, 0.4]$) and the blue curve when the parameter set only contains the AI state ($\eta \in [1.5, 3]$, $g \in [4.5, 6]$ and $J \in [0.1, 0.25]$). The purple line gives the estimation error of the CNN trained for the full parameter set, but evaluated on the restricted parameter set containing the AI state only. To compare the full parameter data set and the AI-only data set, they were both scaled to the range of the full parameter set. Table 7 shows the bias and standard deviation for each of the data sets and estimated parameters. B, Cumulative error distributions, the proportion of absolute errors that fall below a given value, also with all parameters rescaled to $[0, 1]$. This can be understood as the fraction of the data points which are reconstructed better than a specific error. The dashed black lines indicate the 90% coverage interval.

<https://doi.org/10.1371/journal.pcbi.1007725.g007>

g ($g < 4$) which corresponds to the SR state of the network model [28]. The estimation performance of J (panel C) was instead worse for the smallest values of η , that is, in and around the region of parameters where the network model is in an SI state. The estimation of g was generally very accurate for all states of the network (panel B of Fig 8). Taken together this implies that the highest prediction accuracy of the three network parameters is obtained for the AI state.

We next considered the estimation accuracies across the restricted parameter space corresponding to the AI network state only ($\eta \in [1.5, 3]$, $g \in [4.5, 6]$ and $J \in [0.1, 0.25]$),

Table 7. Bias and standard deviation (std) of estimators of network parameters. The table shows the bias and width (std) of the estimator distributions in Fig 7. Note that the biases are all close to zero, indicating that our estimator is essentially unbiased.

Mean and standard deviation of estimators				
		η	g	J
Full parameter	bias	-0.0030	0.0010	0.0049
	std	0.041	0.018	0.042
AI state only	bias	-0.0001	0.0014	0.0009
	std	0.014	0.007	0.024
Full parameter evaluated on AI state only	bias	-0.0080	-0.0006	0.0034
	std	0.017	0.010	0.029

<https://doi.org/10.1371/journal.pcbi.1007725.t007>

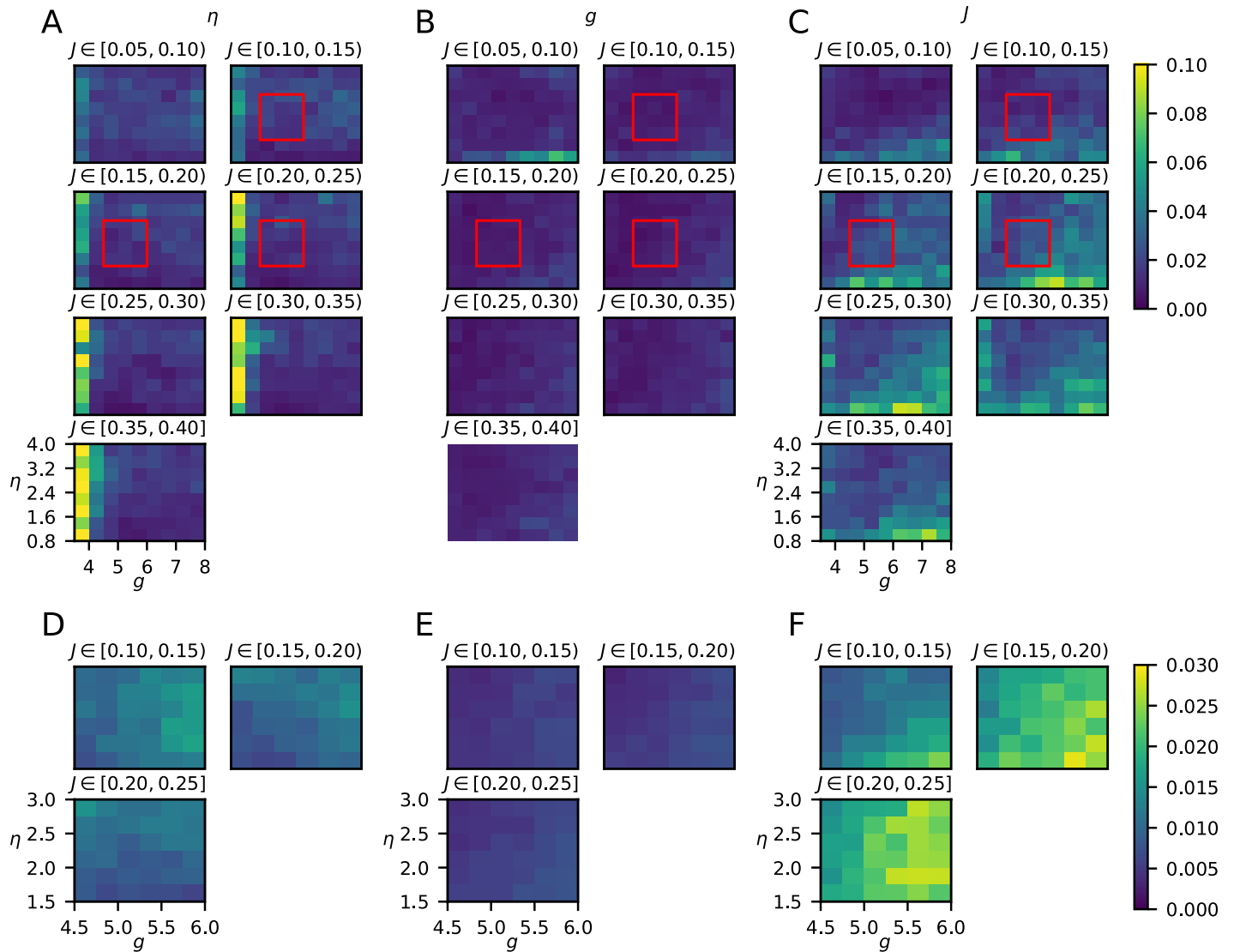


Fig 8. Mean absolute prediction errors. Each voxel in the panels shows the error on the test dataset averaged across the parameter ranges, defined by the pixel size of the grid and the value of J indicated above. A–C, the network trained on the full parameter space. D–F, the network trained on the restricted parameter space. The region of this parameter space is indicated by the red boxes in panels A–C.

<https://doi.org/10.1371/journal.pcbi.1007725.g008>

see Fig 8D–8F. Also within the AI state, g was predicted with the highest accuracy, and J had the lowest estimation accuracy. Further, while the estimation accuracy of g and η was almost constant across the restricted parameter space, the estimation of J became worse with increasing values of J and g (panel F of Fig 8).

Predicting all parameters at once almost as good as using individually trained CNNs

In the above application all three network parameters were predicted by a single convolutional neural net (CNN). We next investigated to what extent the estimation accuracy changed when CNNs were trained to estimate each parameter separately. The results when considering the full parameter space are shown in Fig 9. As expected the estimation accuracy was always better for these ‘single-prediction’ CNN networks: The error distribution of the η prediction was

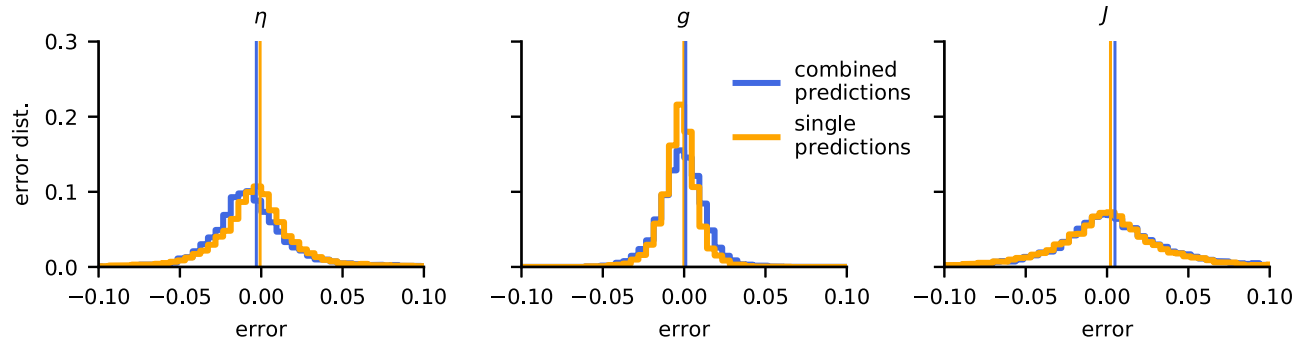


Fig 9. Parameter estimation errors for a single versus multiple CNNs. Comparison of the parameter estimation error orange, when (i) a single CNN is trained to optimise all three parameters η , g and J simultaneously (combined predictions) orange, with (ii) three CNNs each trained to estimate a single parameter (single predictions). All parameters were rescaled to the interval [0, 1]. The bias and standard deviation of the estimators are listed in Table 8.

<https://doi.org/10.1371/journal.pcbi.1007725.g009>

more centered, that is, less biased, for a single prediction network, compared to the ‘combined-prediction’ network (left panel). For the estimation of g , the single-prediction network displayed a narrower peak, also highlighting a slightly better performance. For J , the two approaches gave very similar results. Overall, we conclude that merely small gains are achieved for the present application in terms of estimation accuracy by training a separate CNN for each of the three network parameters.

Randomly sampled training-data preferable

The above estimations were based on CNNs trained by LFPs with random network parameters drawn from uniform distributions. To test if the way the parameter space was sampled had an effect on the accuracy of the estimator, we also generated the same amount of training data on a regular grid, spanning the same parameter space and repeated the training. The estimation accuracy was then computed using a randomly generated test data set, and results are shown in Fig 10. The bias and standard deviation of the estimators are listed in Table 8.

For the prediction of η , there was almost no difference in performance between the CNNs trained with grid-sampled and randomly sampled data (left panel in Fig 10). For g , however, the grid-trained data showed a substantial bias towards lower values of g (middle panel). Such a bias was also seen in the estimation of J , but not so pronounced (right panel).

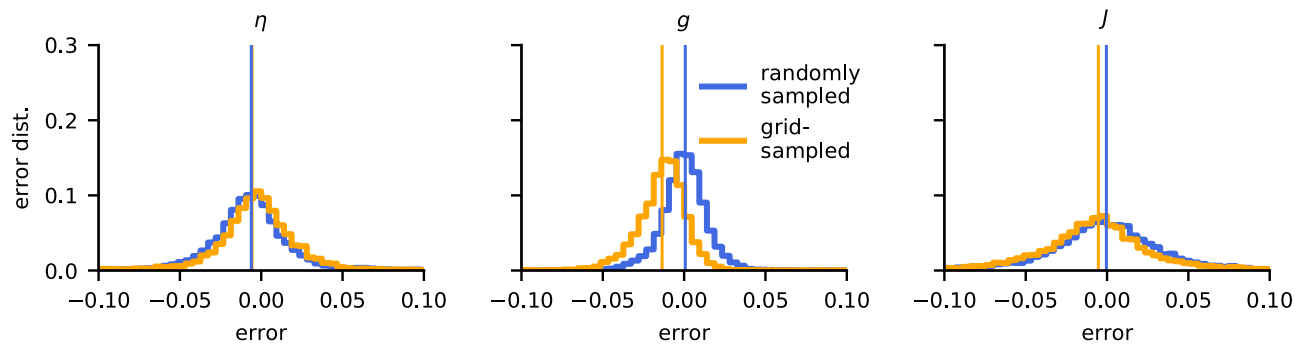


Fig 10. Grid-sampled vs. randomly sampled training data. The plots show error distributions for CNNs trained on data randomly sampled from the parameter set (blue) and from the same amount of training data taken from a regular grid (yellow). All parameters were rescaled to the interval [0, 1]. The bias and standard deviation of the estimators are listed in Table 8.

<https://doi.org/10.1371/journal.pcbi.1007725.g010>

Table 8. Bias and standard deviation (std) of estimators of network parameters. The table shows the bias and width (std) of the estimator distributions in Figs 9 and 10 for single and multi-variable prediction and two different sampling methods. Note that the biases are all close to zero, indicating that our estimator is essentially unbiased.

Mean and standard deviation of estimators				
		η	g	J
Default (random sample, combined prediction)	bias	-0.0030	0.0010	0.0049
	std	0.041	0.018	0.042
Single parameter prediction	bias	-0.0006	-0.0002	0.0021
	std	0.040	0.014	0.043
Grid sample	bias	-0.0056	-0.0136	-0.0053
	std	0.090	0.045	0.052

<https://doi.org/10.1371/journal.pcbi.1007725.t008>

We speculate that training on grid-sampled data introduces a certain lower resolution to the CNN estimators. Randomly sampled data does not contain such a grid scale and eventually (with sufficient training data) enables the network to learn to interpolate on arbitrarily small scales. This intrinsic scale of the grid data might thus be the explanation for the poorer performance of the CNN trained with randomly sampled data.

For this study we only used two simple sampling techniques, grid sampling and random sampling, which were still feasible for the case of only three parameters. However, for further improvement, it might be beneficial to change to more advanced and better performing sampling methods like Latin hypercube sampling [48].

Robustness of parameter estimation

In the above testing we found that the three network parameters η , g , and J generally could be excellently predicted by the LFP signal in the background state. However, in this testing, the other model parameters, that is, the synaptic delay t_d and the parameters describing the neuron dynamics (cf. Table 1), are kept identical to the values used in the training of the CNN. Here we explore the robustness of this performance when this is no longer the case, that is, how does the accuracy of the CNN estimation of synaptic weight parameters change when these other parameters are different in the training and test data sets, mimicking an uncertainty on the true parameter at the stage of training the estimator.

In the example in Fig 11 we show in panels A–C how the estimation errors increase when the synaptic delay t_d is no longer constant in the network, but rather has a Gaussian distribution about the fixed value $t_d = 1.5$ ms used in the training of the CNN. As expected the estimation errors increase with increasing spread σ of the Gaussian. When only the AI states are considered, the estimation error remains small (less than 0.1) even for very large values of the spread where t_d is essentially uniformly distributed between 0 and 3 ms (see panel G). When the full parameter space is considered, the errors remain smaller than 0.1 for values of σ less than 0.1. For larger values of σ , the error increases, mostly so for η and least so for g .

The corresponding results when the membrane time constant τ_m is spread around the single value $\tau_m = 20$ ms used in the training phase, are shown in Fig 11D–11F. Here the estimation errors are very low when only the AI state is considered, generally much smaller than 0.1, even in the extreme case when τ_m is almost uniformly spread between 0 and 40 ms. While the estimation errors increase when the full parameter space is considered, the errors remains smaller than ~ 0.1 when predicting g and J (panels E, F). For η , the prediction errors gets much larger for large values of σ though (panel D).

Also with the firing threshold θ spread around the fixed training-phase value $\theta = 20$ mV (panels G–I), small errors are seen for σ less than 0.1, although larger errors are seen in the

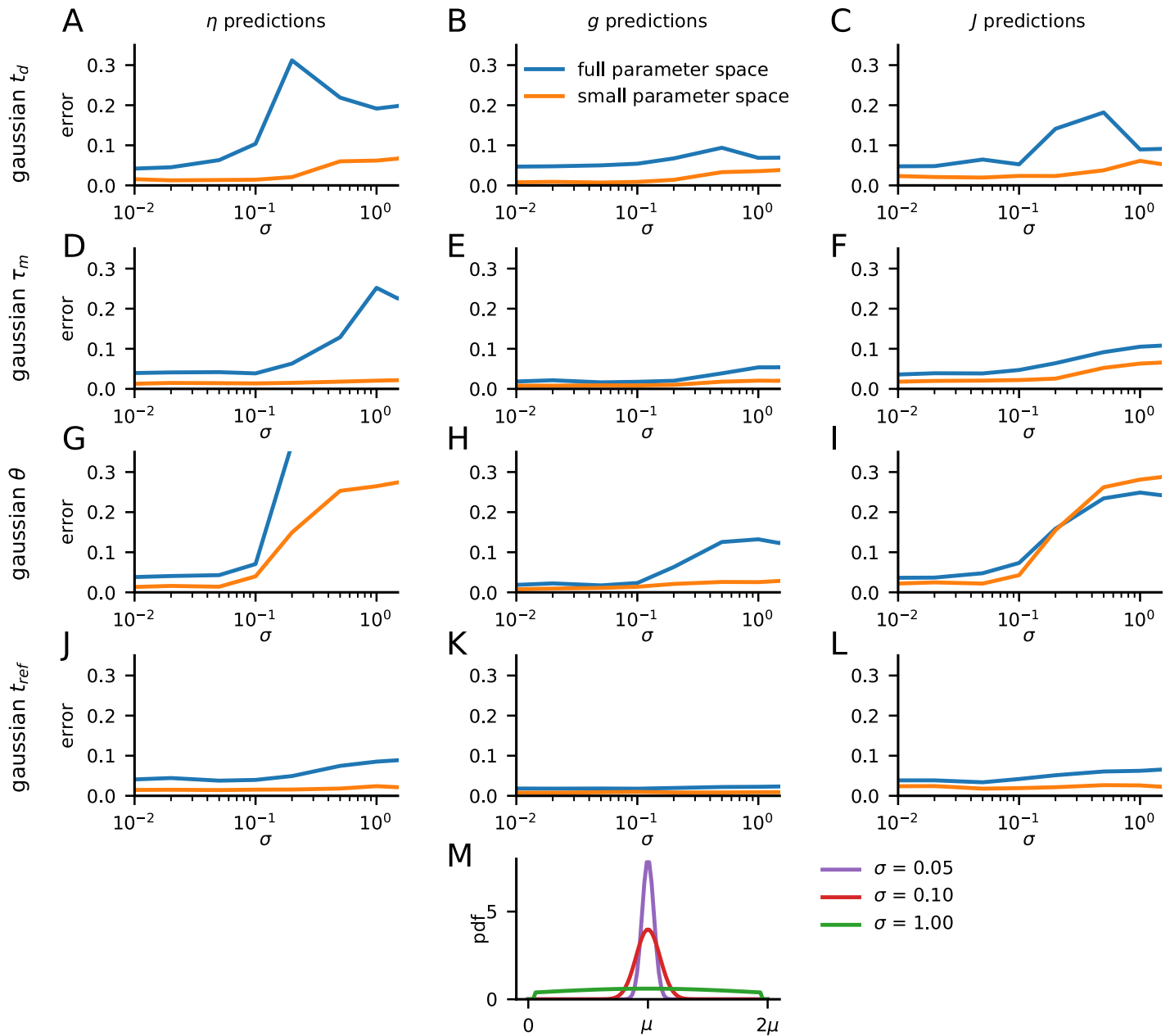


Fig 11. Robustness of estimates with Gaussian spread of model parameters in test data. A–C, estimation errors for η , g , and J , respectively, when the synaptic time delay t_d is randomly distributed when generating test data LFPs. t_d has a truncated Gaussian distribution around the fixed value $t_d = 1.5$ ms used when generating training data, and results for different values of the standard deviation σ are shown (note logarithmic scale). The Gaussian distributions are truncated at 0.2 and 2.8 ms. Results for the average estimation errors across both the full parameter space and the restricted AI parameter spaces are shown. D–F, same as A–C when the neuron membrane time constant τ_m instead is randomly distributed around the training-data value $\tau_m = 20$ ms when generating test data LFPs. The Gaussian distributions are truncated at 2 and 38 ms. G–I, same as A–C when the neuronal firing threshold θ instead is randomly distributed around the training-data value $\theta = 20$ mV when generating test data LFPs. The Gaussian distributions are truncated at 12 and 28 mV. J–L, same as A–C when the refractory period t_{ref} instead is randomly distributed around the training-data value $t_{ref} = 2.0$ ms when generating test data LFPs. The Gaussian distributions are truncated at 0.2 and 3.8 ms. M, Illustration of probability density function (pdf) of truncated Gaussian parameter distributions used in panels A–L.

<https://doi.org/10.1371/journal.pcbi.1007725.g011>

prediction of η and J for larger values of σ . Note that it follows from IF-model as described in Table 1 that with all other parameters fixed, the network dynamics will only depend on the ratio between J and θ . Thus the observed effect of the spread in θ would be directly related to the effect of spread in J in the test data.

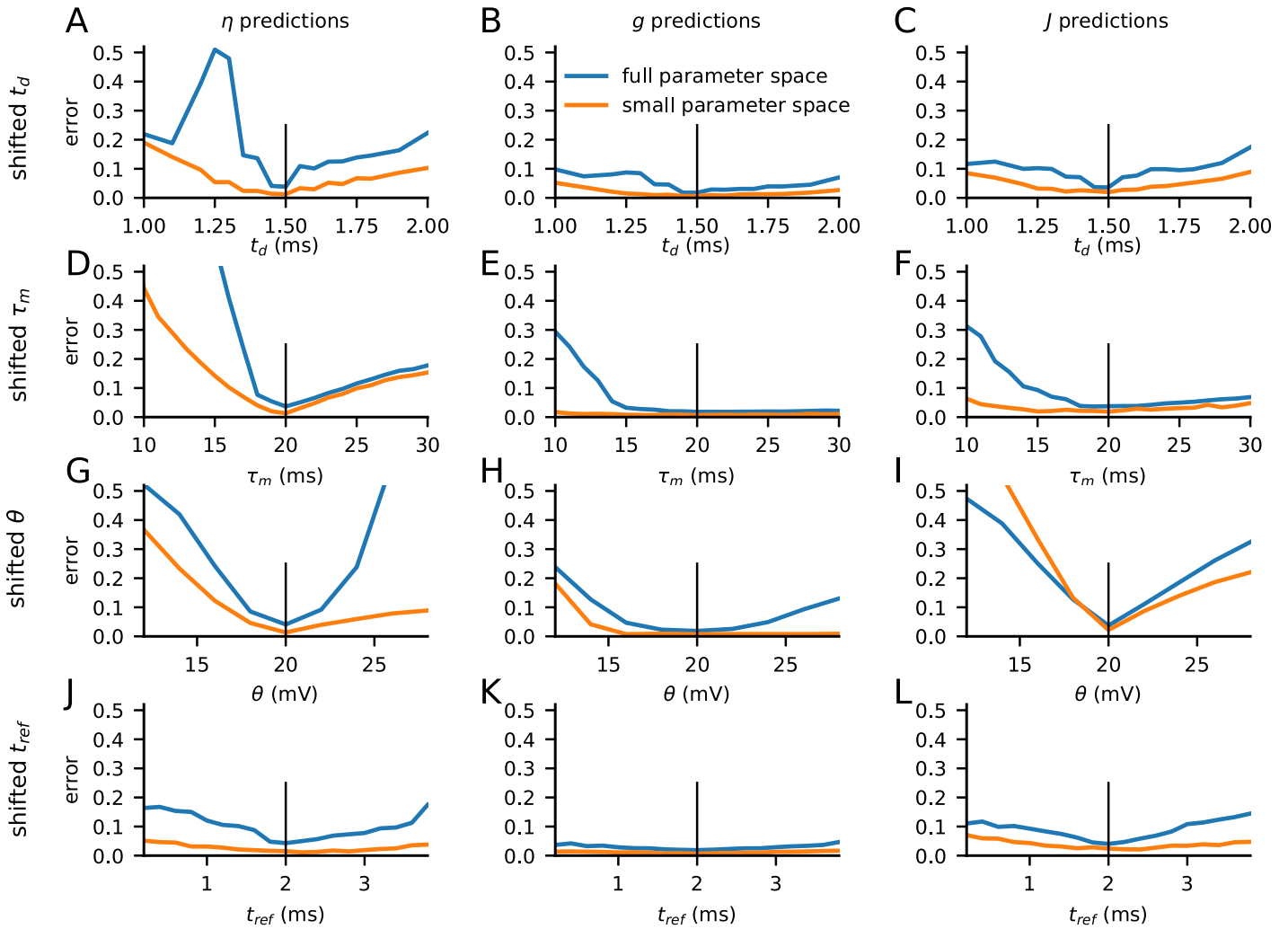


Fig 12. Robustness of estimates with shifts of model parameters in test data. A–C, estimation errors for η , g , and J , respectively, when the synaptic time delay t_d is shifted when generating test data LFPs. $t_d = 1.5$ ms is used when generating training data. Result for the average estimation errors across both the full parameter space and the restricted AI parameter spaces are shown. D–F, same as A–C when the neuron membrane time constant τ_m instead is different than the value $\tau_m = 20$ ms used for generating training data. The vertical bars indicate the x -value used for generating the training data. G–I, same as A–C when the neuronal firing threshold θ instead is different from the value $\theta = 20$ mV used for generating training data. J–L, same as A–C when the refractory period t_{ref} instead is different from the value $t_{ref} = 2.0$ ms used for generating training data.

<https://doi.org/10.1371/journal.pcbi.1007725.g012>

The effect on the estimation errors from the spread of the value of the refractory period around the fixed training-phase value $t_{ref} = 2.0$ ms is seen to be generally small, however (panels J–L).

While the case with heterogeneity in model parameters likely better mimics the situation in brain networks, we also explored in Fig 12 the estimation errors for the situation where the synaptic delay t_d (panels A–C), membrane time constant τ_m (panels D–F), firing threshold θ (panels G–I), and refractory period t_{ref} (panels J–L) are fixed at different values when generating the test data compared to the training data. The smallest errors are found in the estimation of g and J . An exception is the estimation of J with shifted θ (panel I). Here the error is not only generally larger, but the error is also larger for the AI case alone than when the full parameter space is considered. The accuracy in the predictions of η is in general less than for g and J , and

the errors are particularly large for shifted values of t_d , τ_m and θ . In general, the estimation errors always remain low when shifting t_{ref} .

We did not explore in detail why, for example, the prediction of η in general is less robust to differences of t_d , τ_m and θ in the training and test data. However, it is not unexpected that the error in general is larger when the full parameter space is considered rather than only the AI-part of the parameter space. For example, in the SR state the detailed frequency position of the dominant peak in the power spectra (Fig 5) is very sensitive to the values of the synaptic delay parameter t_d . If so, this will increase the estimation error for the full parameter space compared to for the AI part alone.

Discussion

In the present work we have investigated to what extent the local field potential (LFP), that is, the low-frequency part of an extracellular electrical signal, can be used to extract information about synaptic connection weights and external inputs in the underlying network. As a model we considered the well-known and thoroughly analysed Brunel network comprising an excitatory and an inhibitory population of recurrently connected integrate-and-fire (LIF) neurons [28]. Here only three parameters, η , J , and g , describe the strength of the recurrent synaptic connections and the strength of the external synaptic drive: Despite its simplicity, this model exhibits a high diversity of network dynamics, that is, regular or irregular spiking patterns of individual neurons and synchronous or asynchronous spiking across populations.

The LFP generated by the network was computed using a hybrid scheme [29]: Spikes computed by the point-neuron Brunel network were replayed as presynaptic spikes onto biophysically detailed multicompartmental neuron models to compute the LFP as predicted by volume-conductor theory [31, 32]. We then assessed how well the values of the three network parameters could be estimated from the power spectrum of the stationary 'background' LFP signal by application of a convolutional neural net (CNN) [36] and indeed found that all parameters in general could be very accurately estimated. This was the case even when LFPs stemmed from networks in different dynamic states (Figs 7 and 8), but even more so when the LFPs stemmed from the asynchronous irregular (AI) state only (Fig 8). The accuracy of the estimation of the three network parameters was found to be particularly high when the same values of the remaining model parameters (synaptic delay and parameters describing the neurons) were used for the training data and test data. However, the estimates were generally also quite accurate when these parameters were different in the test and training data (Figs 11 and 12).

For the present situation we found that the necessary information for predicting network parameters was contained in the power spectrum (PSD) of the LFP of the background state. While of course this information is also available in the time domain, we found that it was easier to train the CNN using the power spectrum, particularly due to reduced overfitting to the training data. Using Welch's method, the averaging over frequency windows likely reduces the seed-specific variations in the LFPs, leaving less information to overfit to. It is possible that the information contained in the full time series can provide increased accuracy, but it would require more complex networks and possibly much more training data to overcome the issues of overfitting, so we found that in practice using the PSD gave the best results.

Generalization to more complex network models

An obvious question is whether the present successful estimation of network parameters from LFPs will extend to more complex network models with more than three network parameters

like in the Brunel network. Of particular interest here are multilayered cortical network models where several neuronal populations contribute to the LFP signal [24, 29, 49–51].

The estimation problem will likely become more difficult as the number of parameters to estimate increases. However, in the present application we only used the power-spectral density (PSD) of the LFP signals from the stationary background state in the parameter estimation. A ‘richer’ LFP signal which may separate the LFP signals for different parameters better, can be obtained by also including the phase information of the LFP Fourier components, but maybe more importantly by also using stimulus-evoked transient LFP signals. Further, in the present application, the parameters were estimated by LFPs from six channels spanning a depth of 0.5 mm. With only a single population contributing to the LFP as in the present case, fewer channels would in fact have sufficed. When several cortical neuronal populations positioned at different depths contribute to the LFP, the spatial variation of the signal contains more information on the network activity. Here the use of a larger number of channels, spanning all cortical layers, should expectedly improve parameter estimation.

In the present study the network-parameter estimation was found to be particularly successful when the same values of the other model parameters were used for the training and test data. However, the estimation accuracy was generally quite good when these other parameters were different in the test and training data (Figs 11 and 12). This accuracy could likely have been further improved by use of more extensive training data, for example, including LFPs from models with a wider spread of these other parameters in the training data. However, we did not pursue this here.

In the presently studied Brunel network model [28] the external spiking inputs (parameterised by the variable η) were assumed to be stationary and Poissonian. In real networks this external input will generally be more patterned, and a future study could be to investigate how well network parameters can be estimated from LFPs by machine learning techniques in this setting. Further, in the Brunel network the three network parameters are fixed to a single value. In other more comprehensive network models such as the Potjans-Diesmann model [15], however, the network weights are instead described by statistical distributions, rather than individual weights. An interesting line of inquiry would be to explore the use of the present approach to attempt to estimate such distribution parameters from LFPs.

The exact number of unknown model parameters that can be estimated from LFPs alone in different situations, is difficult to predict a priori. In the present case for the Brunel network we observed that the three network parameters could be well estimated from the PSD in the stationary background state alone. For more complex networks described by more parameters, one can imagine using temporally resolved stimulus-evoked LFPs in addition. Combined with the background LFP one would expect that this reduces the above-mentioned ‘redundancy effect’ and allows for ‘disambiguation’ of more candidate network models differing in their parameter values. Even more disambiguation should be possible if the LFPs are supplemented by simultaneously measured spikes and both signals are used as input for CNNs or other machine-learning tools.

Another direction would be to explore the effects of active dendritic conductances such as the I_h channel [52] on parameter estimatability.

Computation of LFPs

To compute the three-second long LFP signals 50000 times to train and test the CNNs in the present study, it was computationally unfeasible to explicitly sum over LFP contributions from each individual presynaptic neuron. Instead we used the approximate formula in Eq 10 based on population firing rates to compute the LFPs, reducing the required computer time by

several orders of magnitude. The accuracy of this approximation for the present network was demonstrated for a set of representative examples (Fig 5). In [29] where the eight-population Potjans-Diesmann [15] cortical network model was considered, the same approximation was seen to give fairly accurate LFPs as well [29, Fig. 13], although not as accurate as in the present case as judged by the example tests. Thus the use of the approximation in Eq 10 to compute the LFPs in future applications should be tested on a case-to-case basis.

Choice of machine-learning method

The choice of using convolutional neural networks (CNNs) within the Keras framework [41] for doing the parameter estimation was made out of convenience. Other machine learning techniques, see [53] for a recent review, could likely have done equally well, or even better. Further, the architecture of the CNNs was not optimized in any systematic way. A systematic study of the best machine learning method to use for LFP-based parameter estimation for more complex network models should be pursued, but is beyond the scope of the present paper.

Implications for analysis of LFPs

For single neurons, biophysics-based modeling is well established [1, 4, 5] and numerous biophysically detailed models with anatomically reconstructed dendrites have been made by fitting to experimental data, for example, [6–8, 10]. These models have mainly been fitted to intracellular electrical recordings, but extracellular recordings [54] and calcium concentrations [55] can also be used.

Until now the analysis of LFPs has largely been based on statistical methods [20, 21]. An overall goal of the present project is to contribute to the investigation of to what extent LFPs also can be used to develop and validate network models in layered brain structures such as cortex and hippocampus. Spikes have already been used to distinguish candidate network models in cortex [18, 56], and LFPs recorded *in vitro* have been used to fit hippocampal network models [27]. There is expectedly a clear link between the accuracy with which a parameter can be (i) estimated from and (ii) fitted to LFP signals. Thus the present observation that network parameters for the Brunel network can be accurately estimated from the background LFPs suggests that the same LFP signal also could be used to accurately fit the same network parameters given that the model structure was known *a priori*. This link between ‘estimability’ and ‘fitability’ should be properly investigated, not only for the Brunel model, but also for more complex network models. However, such a study is beyond the present scope. A related question that also should be investigated is to what extent LFPs can be used to distinguish between candidate models with a different network structure.

Outlook

The recording of single-unit and multi-unit activity (MUA) from the high-frequency part of the extracellular potential, has historically been the most important method for studying *in vivo* activity in neurons and neural networks. However, the interest in the low-frequency part, the LFP, has seen a resurgence in the last decades. One key reason is the development of new multicontact electrodes allowing for high-density electrical recordings across laminae and areas (as well as computers and hard drives allowing for the storage and analysis of the LFP signals). Another reason is the realisation that the LFP offers a unique window into how the dendrites of neurons integrate synaptic inputs for populations of thousands or more neurons [33]. In contrast, the MUA measures the output resulting from this dendritic integration, that is, spikes from a handful of neurons around the electrode contact [57]. Thus spikes and LFPs

offer complementary information about network activity. Since both signals are produced from the same network model, the combined use of spikes and LFPs appears particularly promising for estimation of network model parameters, or for assessing the merit of candidate network models. Such combined use of spikes and LFPs has been shown to be beneficial in identifying laminar neural populations and their synaptic connectivity patterns from multi-electrode cortical recordings [51, 58]. Thus combined use of spikes and LFPs in the estimation of model parameters should be explored in projects with more complex network models where, unlike for the presently considered Brunel network model, the LFP signal is insufficient to alone allow for accurate parameter estimation.

Further, many new optical techniques for probing cortical activity have also been developed and refined, for example, two-photon calcium imaging [59], and voltage-sensitive dye imaging (VSDI), measuring population-averaged membrane potentials [60]. Further, at the systems level one has methods such as electroencephalography (EEG) [61], which measures electrical potentials at the scalp, and magnetoencephalography (MEG) [62] which measures the magnetic field outside the head. These measures can be computed from the activity of candidate network models [63], and tools to facilitate this have been developed [31, 32, 64, 65]. They can all be used to constrain and validate candidate network models, and used in combination they will likely be particularly powerful.

Author Contributions

Conceptualization: Jan-Eirik W. Skaar, Alexander J. Stasik, Espen Hagen, Torbjørn V. Ness, Gaute T. Einevoll.

Formal analysis: Jan-Eirik W. Skaar, Alexander J. Stasik, Espen Hagen, Torbjørn V. Ness, Gaute T. Einevoll.

Funding acquisition: Gaute T. Einevoll.

Investigation: Jan-Eirik W. Skaar, Alexander J. Stasik, Espen Hagen, Torbjørn V. Ness, Gaute T. Einevoll.

Methodology: Jan-Eirik W. Skaar, Alexander J. Stasik, Espen Hagen, Torbjørn V. Ness, Gaute T. Einevoll.

Project administration: Gaute T. Einevoll.

Software: Jan-Eirik W. Skaar, Alexander J. Stasik, Espen Hagen, Torbjørn V. Ness.

Writing – original draft: Jan-Eirik W. Skaar, Alexander J. Stasik, Espen Hagen, Gaute T. Einevoll.

Writing – review & editing: Jan-Eirik W. Skaar, Alexander J. Stasik, Espen Hagen, Torbjørn V. Ness, Gaute T. Einevoll.

References

1. Dayan P, Abbott LF. Theoretical neuroscience. MIT Press, Cambridge; 2001.
2. Einevoll GT, Destexhe A, Diesmann M, Grün S, Jirsa V, de Kamps M, et al. The Scientific Case for Brain Simulations. *Neuron*. 2019; 102:735–744. <https://doi.org/10.1016/j.neuron.2019.03.027> PMID: 31121126
3. Hodgkin AL, Huxley AF. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*. 1952; 117:500–544. <https://doi.org/10.1113/jphysiol.1952.sp004764> PMID: 12991237
4. Koch C. Biophysics of Computation. Oxford Univ Press, Oxford; 1999.

5. Sterratt D, Graham B, Gillies A, Willshaw D. Principles of computational modelling in neuroscience. Cambridge University Press; 2011.
6. Hay E, Hill S, Schürmann F, Markram H, Segev I. Models of Neocortical Layer 5b Pyramidal Cells Capturing a Wide Range of Dendritic and Perisomatic Active Properties. *PLoS Computational Biology*. 2011; 7(7):e1002107. <https://doi.org/10.1371/journal.pcbi.1002107> PMID: 21829333
7. Markram H, Muller E, Ramaswamy S, Reimann MW, Abdellah M, Sanchez CA, et al. Reconstruction and Simulation of Neocortical Microcircuitry. *Cell*. 2015; 163:456–492. <https://doi.org/10.1016/j.cell.2015.09.029> PMID: 26451489
8. Migliore M, Cook EP, Jaffe DB, Turner DA, Johnston D. Computer simulations of morphologically reconstructed CA3 hippocampal neurons. *Journal of Neurophysiology*. 1995; 73(3):1157–1168. <https://doi.org/10.1152/jn.1995.73.3.1157> PMID: 7608762
9. McCormick DA, Huguenard JR. A model of the electrophysiological properties of thalamocortical relay neurons. *J Neurophysiol*. 1992; 68(4):1384–1400. <https://doi.org/10.1152/jn.1992.68.4.1384> PMID: 1331356
10. Halnes G, Augustinaite S, Heggelund P, Einevoll GT, Migliore M. A Multi-Compartment Model for Interneurons in the Dorsal Lateral Geniculate Nucleus. *PLoS Computational Biology*. 2011; 7(9). <https://doi.org/10.1371/journal.pcbi.1002160> PMID: 21980270
11. Jolivet R, Schürmann F, Berger TK, Naud R, Gerstner W, Roth A. The quantitative single-neuron modeling competition. *Biol Cybern*. 2008; 99(4-5):417–426. <https://doi.org/10.1007/s00422-008-0261-x> PMID: 19011928
12. Pozzorini C, Mensi S, Hagens O, Naud R, Koch C, Gerstner W. Automated High-Throughput Characterization of Single Neurons by Means of Simplified Spiking Models. *PLoS Comput Biol*. 2015; 11(6): e1004275. <https://doi.org/10.1371/journal.pcbi.1004275> PMID: 26083597
13. Marder E, Goaillard JM. Variability, compensation and homeostasis in neuron and network function. *Nature Reviews Neuroscience*. 2006; 7(7):563–574. <https://doi.org/10.1038/nrn1949> PMID: 16791145
14. Traub RD, Contreras D, Cunningham MO, Murray H, LeBeau FEN, Roopun A, et al. Single-column thalamocortical network model exhibiting gamma oscillations, sleep spindles, and epileptogenic bursts. *Journal of neurophysiology*. 2005; 93(4):2194–232. <https://doi.org/10.1152/jn.00983.2004> PMID: 15525801
15. Potjans TC, Diesmann M. The Cell-Type Specific Cortical Microcircuit: Relating Structure and Activity in a Full-Scale Spiking Network Model. *Cerebral Cortex*. 2014; 24(3):785–806. <https://doi.org/10.1093/cercor/bhs358> PMID: 23203991
16. Arkhipov A, Gouwens NW, Billeh YN, Gratiy S, Iyer R, Wei Z, et al. Visual physiology of the layer 4 cortical circuit in silico. *PLoS Computational Biology*. 2018; 14(11):e1006535. <https://doi.org/10.1371/journal.pcbi.1006535> PMID: 30419013
17. Billeh YN, Cai B, Gratiy SL, Dai K, Iyer R, Gouwens NW, et al. Systematic Integration of Structural and Functional Data into Multi-Scale Models of Mouse Primary Visual Cortex. *Neuron*, 2020. <https://doi.org/10.1016/j.neuron.2020.01.040>
18. Blomquist P, Devor A, Indahl UG, Ulbert I, Einevoll GT, Dale AM. Estimation of thalamocortical and intracortical network models from joint thalamic single-electrode and cortical laminar-electrode recordings in the rat barrel system. *PLoS Computational Biology*. 2009; 5(3):e1000328. <https://doi.org/10.1371/journal.pcbi.1000328> PMID: 19325875
19. Buzsáki G, Anastassiou C, Koch C. The origin of extracellular fields and currents—EEG, ECoG, LFP and spikes. *Nature Reviews Neuroscience*. 2012; 13(6):407–420. <https://doi.org/10.1038/nrn3241> PMID: 22595786
20. Einevoll GT, Kayser C, Logothetis NK, Panzeri S. Modelling and analysis of local field potentials for studying the function of cortical circuits. *Nature Reviews Neuroscience*. 2013; 14(11):770–785. <https://doi.org/10.1038/nrn3599> PMID: 24135696
21. Pesaran B, Vinck M, Einevoll GT, Sirota A, Fries P, Siegel M, et al. Investigating large-scale brain dynamics using field potential recordings: analysis and interpretation. *Nature Neuroscience*. 2018; 21(7):903–919. <https://doi.org/10.1038/s41593-018-0171-8> PMID: 29942039
22. Mazzoni A, Panzeri S, Logothetis NK, Brunel N. Encoding of naturalistic stimuli by local field potential spectra in networks of excitatory and inhibitory neurons. *PLoS Computational Biology*. 2008; 4(12): e1000239. <https://doi.org/10.1371/journal.pcbi.1000239> PMID: 19079571
23. Mazzoni A, Brunel N, Cavallari S, Logothetis NK, Panzeri S. Cortical dynamics during naturalistic sensory stimulations: experiments and models. *Journal of Physiology, Paris*. 2011; 105(1-3):2–15. <https://doi.org/10.1016/j.jphysparis.2011.07.014> PMID: 21907800
24. Reimann MW, Anastassiou CA, Perin R, Hill SL, Markram H, Koch C. A biophysically detailed model of neocortical local field potentials predicts the critical role of active membrane currents. *Neuron*. 2013; 79(2):375–390. <https://doi.org/10.1016/j.neuron.2013.05.023> PMID: 23889937

25. Sanjay M, Neymotin SA, Krothapalli SB. Impaired dendritic inhibition leads to epileptic activity in a computer model of CA3. *Hippocampus*. 2015; 25:1336–1350. <https://doi.org/10.1002/hipo.22440> PMID: 25864919
26. Bezaire MJ, Raikov I, Burk K, Vyas D, Soltesz I. Interneuronal mechanisms of hippocampal theta oscillations in a full-scale model of the rodent CA1 circuit. *eLife*. 2016; 5. <https://doi.org/10.7554/eLife.18566> PMID: 28009257
27. Chatzikalymniou AP, Skinner FK. Deciphering the Contribution of Oriens-Lacunosum/Moleculare (OLM) Cells to Intrinsic Theta Rhythms Using Biophysical Local Field Potential (LFP) Models. *eNeuro*. 2018; 5. <https://doi.org/10.1523/ENEURO.0146-18.2018> PMID: 30225351
28. Brunel N. Dynamics of sparsely connected networks of excitatory and inhibitory neurons. *Computational Neuroscience*. 2000; 8:183–208. <https://doi.org/10.1023/A:1008925309027>
29. Hagen E, Dahmen D, Stavrinou ML, Lindén H, Tetzlaff T, van Albada SJ, et al. Hybrid Scheme for Modeling Local Field Potentials from Point-Neuron Networks. *Cerebral Cortex*. 2016; 26(12):4461–4496. <https://doi.org/10.1093/cercor/bhw237> PMID: 27797828
30. Kunkel S, Morrison A, Weidel P, Eppler JM, Sinha A, Schenck W, et al. NEST 2.12.0; 2017. Available from: <https://doi.org/10.5281/zenodo.259534>.
31. Lindén H, Hagen E, Łęski S, Norheim ES, Pettersen KH, Einevoll GT. LFPy: a tool for biophysical simulation of extracellular potentials generated by detailed model neurons. *Frontiers in Neuroinformatics*. 2014; 7(41):1–15.
32. Hagen E, Næss S, Ness TV, Einevoll GT. Multimodal Modeling of Neural Network Activity: Computing LFP, ECoG, EEG, and MEG Signals With LFPy 2.0. *Frontiers in Neuroinformatics*. 2018; 12. <https://doi.org/10.3389/fninf.2018.00092>
33. Lindén H, Tetzlaff T, Potjans TC, Pettersen KH, Grün S, Diesmann M, et al. Modeling the spatial reach of the LFP. *Neuron*. 2011; 72(5):859–872. <https://doi.org/10.1016/j.neuron.2011.11.006> PMID: 22153380
34. Łęski S, Lindén H, Tetzlaff T, Pettersen KH, Einevoll GT. Frequency Dependence of Signal Power and Spatial Reach of the Local Field Potential. *PLoS Computational Biology*. 2013; 9(7):e1003137. <https://doi.org/10.1371/journal.pcbi.1003137> PMID: 23874180
35. Mazzoni A, Lindén H, Cuntz H, Lansner A, Panzeri S, Einevoll GT. Computing the Local Field Potential (LFP) from Integrate-and-Fire Network Models. *PLoS Computational Biology*. 2015; 11:e1004584. <https://doi.org/10.1371/journal.pcbi.1004584> PMID: 26657024
36. Rawat W, Wang Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural computation*. 2017; 29:2352–2449. https://doi.org/10.1162/NECO_a_00990 PMID: 28599112
37. Nordlie E, Gewaltig MO, Plesser HE. Towards Reproducible Descriptions of Neuronal Network Models. *PLoS Computational Biology*. 2009; 5(8):e1000456. <https://doi.org/10.1371/journal.pcbi.1000456> PMID: 19662159
38. Welch P. The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*. 1967; 15(2):70–73. <https://doi.org/10.1109/TAU.1967.1161901>
39. Jones E, Oliphant T, Peterson P, et al. SciPy: Open source scientific tools for Python; 2001–. Available from: <http://www.scipy.org/>.
40. Grün S, Rotter S. Analysis of Parallel Spike Trains. Springer Series in Computational Neuroscience. Springer US; 2010. Available from: <https://books.google.no/books?id=dUoCOZXP2FkC>.
41. Chollet F, et al. Keras. 2015.
42. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015.
43. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. *CoRR*. 2014;abs/1412.6980.
44. Hines ML, Davison AP, Muller E. NEURON and Python. *Frontiers in Neuroinformatics*. 2009; 3(January):1. <https://doi.org/10.3389/neuro.11.001.2009> PMID: 19198661
45. Ecker AS, Berens P, Keliris GA, Bethge M, Logothetis NK, Tolias AS. Decorrelated Neuronal Firing in Cortical Microcircuits. *Science*. 2010; 327(5965):584–587. <https://doi.org/10.1126/science.1179867> PMID: 20110506
46. Mochizuki Y, Onaga T, Shimazaki H, Shimokawa T, Tsubo Y, Kimura R, et al. Similarity in Neuronal Firing Regimes across Mammalian Species. *Journal of Neuroscience*. 2016; 36(21):5736–5747. <https://doi.org/10.1523/JNEUROSCI.0230-16.2016> PMID: 27225764
47. Lindén H, Pettersen KH, Einevoll GT. Intrinsic dendritic filtering gives low-pass power spectra of local field potentials. *Journal of computational neuroscience*. 2010; 29(3):423–444. <https://doi.org/10.1007/s10827-010-0245-4> PMID: 20502952

48. McKay MD, Beckman RJ, Conover WJ. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*. 1979; 21(2):239–245. <https://doi.org/10.1080/00401706.1979.10489755>
49. Głańska H, Potworowski J, Łęski S, Wójcik DK. Independent Components of Neural Activity Carry Information on Individual Populations. *PLoS ONE*. 2014; 9(8):e105071. <https://doi.org/10.1371/journal.pone.0105071> PMID: 25153730
50. Tomsett RJ, Ainsworth M, Thiele A, Sanayei M, Chen X, Gieselmann MA, et al. Virtual Electrode Recording Tool for Extracellular potentials (VERTEX): comparing multi-electrode recordings from simulated and biological mammalian cortical tissue. *Brain Structure and Function*. 2015; 220(4):2333–2353. <https://doi.org/10.1007/s00429-014-0793-x> PMID: 24863422
51. Głańska HT, Norheim E, Devor A, Dale AM, Einevoll GT, Wójcik DK. Generalized Laminar Population Analysis (gLPA) for Interpretation of Multielectrode Data from Cortex. *Frontiers in Neuroinformatics*. 2016; 10:1. <https://doi.org/10.3389/fninf.2016.00001> PMID: 26834620
52. Ness TV, Remme MWH, Einevoll GT. h-Type Membrane Current Shapes the Local Field Potential from Populations of Pyramidal Neurons. *The Journal of neuroscience*. 2018; 38:6011–6024. <https://doi.org/10.1523/JNEUROSCI.3278-17.2018> PMID: 29875266
53. Ismail Fawaz H, Forestier G, Weber J, Idoumghar L, Muller PA. Deep learning for time series classification: a review. *arXiv*. 2018; p. arXiv:1809.04356.
54. Gold C, Henze DA, Koch C. Using extracellular action potential recordings to constrain compartmental models. *Journal of Computational Neuroscience*. 2007; 23(1):39–58. <https://doi.org/10.1007/s10827-006-0018-2> PMID: 17273940
55. Mäki-Marttunen T, Halmes G, Devor A, Metzner C, Dale AM, Andreassen OA, et al. A stepwise neuron model fitting procedure designed for recordings with high spatial resolution: Application to layer 5 pyramidal cells. *Journal of Neuroscience Methods*. 2018; 293:264–283. <https://doi.org/10.1016/j.jneumeth.2017.10.007> PMID: 28993204
56. Stimberg M, Wimmer K, Martin R, Schwabe L, Mariño J, Schummers J, et al. The Operating Regime of Local Computations in Primary Visual Cortex. *Cerebral Cortex*. 2009; 19(9):2166–2180. <https://doi.org/10.1093/cercor/bhn240> PMID: 19221143
57. Buzsáki G. Large-scale recording of neuronal ensembles. *Nature Neuroscience*. 2004; 7(5):446–451. <https://doi.org/10.1038/nn1233> PMID: 15114356
58. Einevoll GT, Pettersen KH, Devor A, Ulbert I, Hålgren E, Dale AM. Laminar Population Analysis: Estimating Firing Rates and Evoked Synaptic Activity From Multielectrode Recordings in Rat Barrel Cortex. *Journal of Neurophysiology*. 2007; 97(3):2174–2190. <https://doi.org/10.1152/jn.00845.2006> PMID: 17182911
59. Helmchen F, Denk W. Deep tissue two-photon microscopy. *Nature Methods*. 2005; 2(12):932–940. <https://doi.org/10.1038/nmeth818> PMID: 16299478
60. Grinvald A, Hildesheim R. VSDI: a new era in functional imaging of cortical dynamics. *Nature Reviews Neuroscience*. 2004; 5:874–885. <https://doi.org/10.1038/nrn1536> PMID: 15496865
61. Nunez PL, Srinivasan R. *Electric fields of the brain: The Neurophysics of EEG*. 2nd ed. Oxford University Press, Inc.; 2006.
62. Hämmäläinen M, Hari R, Ilmoniemi RJ, Knuutila J, Lounasmaa OV. Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of Modern Physics*. 1993; 65(2):413–497. <https://doi.org/10.1103/RevModPhys.65.413>
63. Brette R, Destexhe A, editors. *Handbook of Neural Activity Measurement*. Cambridge University Press; 2012.
64. Hagen E, Næss S, Ness TV, Einevoll GT. LFPy—multimodal modeling of extracellular neuronal recordings in Python. In: *Encyclopedia of Computational Neuroscience*. Springer, New York, NY; 2019. p. 620286. Available from: https://doi.org/10.1007/978-1-4614-7320-6_100681-1.
65. Gratiy SL, Billeh YN, Dai K, Mitelut C, Feng D, Gouwens NW, et al. BioNet: A Python interface to NEURON for modeling large-scale networks. *PLoS ONE*. 2018; 13(8):e0201630. <https://doi.org/10.1371/journal.pone.0201630> PMID: 30071069