



OPEN ACCESS

EDITED BY

Alberto Antonietti,
Ecole Polytechnique Fédérale de
Lausanne (EPFL), Switzerland

REVIEWED BY

James Isbister,
Ecole Polytechnique Fédérale de
Lausanne (EPFL), Switzerland
Kyriaki Sidiropoulou,
University of Crete, Greece

*CORRESPONDENCE

Tobias Schulte to Brinke
t.schulte.to.brinke@fz-juelich.de

RECEIVED 19 April 2022

ACCEPTED 01 September 2022

PUBLISHED 03 October 2022

CITATION

Schulte to Brinke T, Duarte R and
Morrison A (2022) Characteristic
columnar connectivity caters to
cortical computation: Replication,
simulation, and evaluation of a
microcircuit model.

Front. Integr. Neurosci. 16:923468.
doi: 10.3389/fnint.2022.923468

COPYRIGHT

© 2022 Schulte to Brinke, Duarte and
Morrison. This is an open-access
article distributed under the terms of
the [Creative Commons Attribution
License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution
or reproduction in other forums is
permitted, provided the original
author(s) and the copyright owner(s)
are credited and that the original
publication in this journal is cited, in
accordance with accepted academic
practice. No use, distribution or
reproduction is permitted which does
not comply with these terms.

Characteristic columnar connectivity caters to cortical computation: Replication, simulation, and evaluation of a microcircuit model

Tobias Schulte to Brinke^{1,2*}, Renato Duarte^{1,3} and
Abigail Morrison^{1,2}

¹Institute of Neuroscience and Medicine (INM-6) and Institute for Advanced Simulation (IAS-6) and JARA-BRAIN Institute I, Jülich Research Centre, Jülich, Germany, ²Department of Computer Science 3 - Software Engineering, RWTH Aachen University, Aachen, Germany, ³Donders Institute for Brain, Cognition and Behavior, Radboud University, Nijmegen, Netherlands

The neocortex, and with it the mammalian brain, achieves a level of computational efficiency like no other existing computational engine. A deeper understanding of its building blocks (cortical microcircuits), and their underlying computational principles is thus of paramount interest. To this end, we need reproducible computational models that can be analyzed, modified, extended and quantitatively compared. In this study, we further that aim by providing a replication of a seminal cortical column model. This model consists of noisy Hodgkin-Huxley neurons connected by dynamic synapses, whose connectivity scheme is based on empirical findings from intracellular recordings. Our analysis confirms the key original finding that the specific, data-based connectivity structure enhances the computational performance compared to a variety of alternatively structured control circuits. For this comparison, we use tasks based on spike patterns and rates that require the systems not only to have simple classification capabilities, but also to retain information over time and to be able to compute nonlinear functions. Going beyond the scope of the original study, we demonstrate that this finding is independent of the complexity of the neuron model, which further strengthens the argument that it is the connectivity which is crucial. Finally, a detailed analysis of the memory capabilities of the circuits reveals a stereotypical memory profile common across all circuit variants. Notably, the circuit with laminar structure does not retain stimulus any longer than any other circuit type. We therefore conclude that the model's computational advantage lies in a sharper representation of the stimuli.

KEYWORDS

neocortex, cortical column, microcircuit, connectivity structure, neuron dynamics, replication, reproduction

1. Introduction

Neurons of the neocortex are arranged in layers, forming connectivity structures through their synapses that share many properties across various brain areas. This suggests that diverse cortical areas are likely based on a common microcircuit template (see e.g., Mountcastle, 1997; Horton and Adams, 2005; DeFelipe, 2012; Harris and Shepherd, 2015). These broad commonalities suggest a functional purpose behind this structure that gives networks an information processing advantage over randomly connected circuits.

To investigate this hypothesis, Stefan Häusler and Wolfgang Maass developed a data-based microcircuit model and tested its computational properties in comparison with networks with equivalent dynamics but alternative connectivity structures in their seminal paper *A Statistical Analysis of Information-Processing Properties of Lamina-Specific Cortical Microcircuit Models* (Häusler and Maass, 2007).

After this paper was published as one of the first studies on data-based cortical column models, it was cited hundreds of times and influenced the computational neuroscience community's view on the purpose and benefits of a laminar cortical network structure. Since then, the model has been used by Wolfgang Maass's team to analyze, for example, the distributions of network motifs in its connectivity structure (Häusler et al., 2009) and to show how a version of the network with stochastic neurons can exploit noise for computation (Habenschuss et al., 2009; Maass, 2014). Rasch et al. (2011) use the model as the basis for a larger network that also includes a model of the retina and lateral geniculate nucleus (LGN) of the thalamus to analyze its responses to natural stimuli and compare them with *in vivo* activity. Since the publication of Häusler and Maass (2007), modeling of cortical columns has partially evolved toward large-scale networks of point neurons whose focus is to accurately reproduce the statistical properties of spike activity from *in vivo* data (e.g., Potjans and Diesmann, 2014). The other recent direction of modeling cortical columns focuses on large networks of biophysically detailed neural compartment models, such as Markram et al. (2015) and Billeh et al. (2020). Nevertheless, smaller network models continue to be relevant because simulating these large-scale models requires huge amounts of computing resources that are beyond the scope of many computational laboratories. Therefore, as a highly influential study uncovering the relationships between structure, dynamics and function, it would be of great benefit to have the computational model introduced by Häusler and Maass (2007) available for further study and quantitative comparison with other models.

Unfortunately, as the model was originally implemented in MATLAB (unknown version, but no later than R2006b) and the C++ simulation plugin *csim* is no longer maintained, the code can no longer be executed. In this article we present a replication

of the original study, which serves the twin purpose of testing the original findings and providing an executable version of the model to the computational neuroscience community. Specifically, we re-implement their model using the open source softwares *NEST* (Hahne et al., 2021) to simulate the networks, *NESTML* (Babu et al., 2021) to define the neuron model and Python for data analysis, thus ensuring a reusable and maintainable code base.

Here, we use the term *replication* in the R⁵ sense described by Benureau and Rougier (2018), i.e., striving to obtain the same results using an independent code base, whereas a *reproduction* (R³) of the model would have been achieved if we had obtained the results of the original study using the original code. Note that others have argued that these terms should be used the other way around: see Plesser (2018) for an overview and analysis.

Following the structure of the original work, we construct a cortical column model based on data from rat and cat cortical areas published by Thomson (2002). The network consists of spiking Hodgkin-Huxley neurons with an intrinsic conductance-based noise mechanism that represents the incoming currents generated by stochastically releasing synapses and is connected by synapses with short-term plasticity. Using this network model, we investigate the impact of the data-based laminar structure on the computational performance of the system. Besides the data-based model, we implement additional control models that share the global statistics of the microcircuit whilst removing specific network properties. This allows analysis of how different network properties affect the networks' computational performance on various tasks based on input signals that are encoded as precise spike patterns or spike trains with changing firing rates.

These tasks are designed in such a way that they allow us to draw conclusions about computational abilities of the models under investigation by testing the networks not only on their simple classification capabilities, but also on memory and their nonlinear processing power. Following the reservoir computing paradigm, the synaptic efficacies of the recurrent connections within the network are not trained to improve performance; only the projections from the network to separate readout neurons are learned.

We successfully reproduce the main data-based model and all six control circuit variants. The results on the computational tasks confirm the findings of the original study, most notably that the data-based circuit has superior computational performance to circuits without laminar structure.

Going beyond the experiments of the original study, and demonstrating the value of having executable versions of important models, we further examine the generalizability of the results with respect to the neuron model. Assuming that the laminar structure is the most important component of the model, we hypothesize that the central findings are not dependent on the specific choice of the somewhat complex

Hodgkin-Huxley neurons used in the original study. To investigate this hypothesis, we simplify the neuron model by reducing its complexity to basic integrate-and-fire dynamics and show that this simplification not only maintains the superior performance of the data-based circuit, but even increases its absolute performance on almost all tasks. The same is true for the removal of the noise mechanism from the Hodgkin-Huxley model. Although noise was added mainly to increase biological plausibility rather than to improve performance, it is not necessarily the case that noise degrades the performance of a neural system, since, for example, effects such as stochastic resonance can improve the detection of weak signals (Wiesenfeld and Moss, 1995; McDonnell and Ward, 2011).

Finally, we extend the original computational tasks to include a more detailed examination of the memory capabilities of the systems under consideration, reflecting the fact that the ability to recall information over time forms the basis for a variety of cognitive processes. Our results reveal a stereotypical memory profile for all tested circuits and demonstrate that the characteristic temporal structure of the stimulus has differential effects on the task performance of the networks receiving it.

Apart from providing a reproducible and re-usable implementation of the cortical microcircuit model in Häusler and Maass (2007), our successful replication reduces the likelihood that the original findings were influenced by implementation errors (Benureau and Rougier, 2018; Pauli et al., 2018). Our findings thus lend further support to the hypothesis that the highly nonrandom connectivity structure of cortical columns serves important computational purposes, with the degree distributions, i.e., the distributions of the number of incoming and outgoing connections per neuron, playing the most prominent role. Going beyond the original findings, we further demonstrate that the computational benefits of the laminar structure are not dependent on the complexity of the neuron model. Finally, we discover that the laminar structure does not confer memory benefits in the model—the circuits with laminar structure do not retain stimulus information for longer than networks with other connectivity assumptions—and conclude that the superior computational performance is achieved primarily by generating more distinct stimulus representations.

2. Materials and methods

2.1. Microcircuit model

In the following sections we provide details of our implementation of the microcircuit model that is publicly available at Zenodo (Schulte to Brinke et al., 2022) and compare it to the model described in Häusler and Maass (2007), whose implementation is available at ModelDB (McDougal et al., 2017; accession number

82385, <https://senselab.med.yale.edu/ModelDB/showmodel?model=82385>).

2.1.1. Neuron model

The networks consist of single-compartment Hodgkin-Huxley type neurons with three different active currents, as described by Destexhe and Paré (1999), and an intrinsic conductance noise mechanism introduced by Destexhe et al. (2001):

$$C_m \frac{dV_m}{dt} = -g_L(V_m - E_L) - I_{Na} - I_K - I_M - I_{noise} \quad (1)$$

where V_m is the membrane potential, C_m is the membrane capacitance, g_L is the leak conductance and E_L is the leak reversal potential. I_{Na} is a voltage-dependent Na^+ current with the following dynamics:

$$I_{Na} = g_{Na} m^3 h (V_m - E_{Na}) \quad (2)$$

$$\frac{dm}{dt} = \alpha_m(V_m)(1 - m) - \beta_m(V_m)m \quad (3)$$

$$\frac{dh}{dt} = \alpha_h(V_m)(1 - h) - \beta_h(V_m)h \quad (4)$$

$$\alpha_m = \frac{-0.32(V_m - V_T - 13)}{\exp[-(V_m - V_T - 13)/4] - 1} \quad (5)$$

$$\beta_m = \frac{0.28(V_m - V_T - 40)}{\exp[(V_m - V_T - 40)/5] - 1} \quad (6)$$

$$\alpha_h = 0.128 \exp[-(V_m - V_T - V_S - 17)/18] \quad (7)$$

$$\beta_h = \frac{4}{1 + \exp[-(V_m - V_T - V_S - 40)/5]} \quad (8)$$

where g_{Na} is the sodium peak conductance, E_{Na} is the sodium reversal potential, V_S is a voltage that shifts the inactivation toward hyperpolarized values and V_T is a voltage offset that controls dynamics and adjusts the membrane threshold. Note, the model does not incorporate an explicit threshold; the membrane potential threshold V_{thresh} in Table 1 is just the potential at which the peak in the membrane potential is recognized as a spike by the simulator. This is also the reason why a refractory period t_{ref} is needed to avoid the emission of multiple spikes during a peak in the membrane potential. I_K is a delayed-rectifier K^+ current:

$$I_K = g_K n^4 (V_m - E_K) \quad (9)$$

$$\frac{dn}{dt} = \alpha_n(V_m)(1 - n) - \beta_n(V_m)n \quad (10)$$

$$\alpha_n = \frac{-0.032(V_m - V_T - 15)}{\exp[-(V_m - V_T - 15)/5] - 1} \quad (11)$$

$$\beta_n = 0.5 \exp[-(V_m - V_T - 10)/40] \quad (12)$$

TABLE 1 Neuron parameters.

Parameter	Value	Source	Description
V_m	uniformly distributed between -70 and -60 mV	Paper	Membrane potential
V_T	-63 (-58) mV	Code (Destexhe and Paré, 1999)	Voltage offset that controls dynamics
V_{thresh}	-30 mV	Code	Membrane potential threshold
V_S	-10 mV	Destexhe and Paré, 1999	Shifting voltage
E_L	-80 mV	Destexhe et al., 2001	Leak reversal potential
a	34,636 μm^2	Paper	Membrane area used for all but g_M
ρ_{C_m}	1 $\mu\text{F}/\text{cm}^2$	Destexhe et al., 2001	Membrane capacitance density
C_m	346.36 pF	$a \cdot \rho_{C_m}$	Capacity of the membrane
ρ_{g_L}	0.045 mS/cm ²	Destexhe et al., 2001	Leak conductance density
g_L	15.5862 nS	$a \cdot \rho_{g_L}$	Leak conductance
$\tau_{syn_{ex}}$	3 ms	Code	Time constant of the excitatory synaptic exponential function
$\tau_{syn_{in}}$	6 ms	Code	Time constant of the inhibitory synaptic exponential function
t_{ref}	3 ms	Code	Duration of refractory period
E_{ex}	0 mV	Destexhe et al., 2001	Excitatory synaptic reversal potential
E_{in}	-75 mV	Destexhe et al., 2001	Inhibitory synaptic reversal potential
E_{Na}	50 (60) mV	Code (Mainen et al., 1995)	Sodium reversal potential
$\rho_{g_{Na}}$	516 (500) pS/ μm^2	Code (paper)	Peak conductance density for I_{Na}
g_{Na}	17,872.176 nS (17,318 nS)	$a \cdot \rho_{g_{Na}}$ Code (paper)	Sodium peak conductance
E_K	-90 mV	Code	Potassium reversal potential
ρ_{g_K}	100 pS/ μm^2	Paper	Peak conductance density for I_K
g_K	3,463.6 nS	$a \cdot \rho_{g_K}$	Potassium peak conductance
a_M	10,000 μm^2	Code	Membrane area used for g_M
E_M	-80 (-90) mV	Code (Mainen et al., 1995)	Potassium reversal potential for I_M
ρ_{g_M}	10 (5) pS/ μm^2	Code (paper)	Peak conductance density for I_M
$g_{M_{ex}}$	100 (173.18) nS	$a_M \cdot \rho_{g_M}$ Code ($a \cdot \rho_{g_M}$ paper)	Peak conductance of additional potassium ion channel in exc. neurons
$g_{M_{in}}$	0 nS	Paper	Peak conductance of additional Potassium ion channel in inh. neurons

The source column indicates where the value can be found, searching in the following order: main replicated paper, referenced papers, source code. If a value was given in the paper which differs from the one used in the code, the paper value is written in parenthesis.

Here, g_K is the potassium peak conductance and E_K is the potassium reversal potential. The third current is a non-inactivating K^+ current responsible for spike frequency adaptation, which is only activated for excitatory neurons and was first described by Mainen et al. (1995):

$$I_M = g_M p (V_m - E_M) \quad (13)$$

$$\frac{dp}{dt} = \alpha_p (V_m) (1 - p) - \beta_p (V_m) p \quad (14)$$

$$\alpha_p = \frac{r \cdot (V_m + 30)}{1 - \exp[-(V_m + 30)/9]} \quad (15)$$

$$\beta_p = \frac{-r \cdot (V_m + 30)}{1 - \exp[-(V_m + 30)/9]} \quad (16)$$

where g_M is the peak conductance for this additional potassium channel. The factor r is set to 0.0001 by Destexhe et al. (2001), but Häusler and Maass (2007) use a value of 0.001 in their code,

and we followed the latter in our implementation. A complete specification of all neuron parameters can be found in Table 1.

In addition to these ion channel dynamics, Häusler and Maass (2007) used noisy background currents whose conductances are modeled by an Ornstein-Uhlenbeck process. This stochastic background activity was introduced by Destexhe et al. (2001) to represent the spontaneous activations of incoming synapses as follows:

$$I_{noise} = g_{ne}(t)(V_m - E_{ex}) + g_{ni}(t)(V_m - E_{in}) \quad (17)$$

where g_{ne} is the time dependent excitatory conductance, E_{ex} is the excitatory synaptic reversal potential and g_{ni} and E_{in} are their inhibitory counterparts.

The conductances are calculated by the following update rules:

$$g_{ne}(t+h) = g_{e0} + [g_{ne}(t) - g_{e0} \exp(-h/\tau_{ne}) + A_e N_1(0, 1)] \quad (18)$$

$$g_{ni}(t+h) = g_{i0} + [g_{ni}(t) - g_{i0} \exp(-h/\tau_{ni}) + A_i N_2(0, 1)] \quad (19)$$

where g_{e0} and g_{i0} are average conductances, τ_{ne} and τ_{ni} are time constants, h is the integration step, $N_1(0, 1)$ and $N_2(0, 1)$ are random numbers drawn from a normal distribution with mean 0 and standard deviation 1, and A_e and A_i are amplitude coefficients given by

$$A_e = \sqrt{\frac{D_e \tau_{ne}}{2} [1 - \exp(\frac{-2h}{\tau_{ne}})]} \quad (20)$$

$$A_i = \sqrt{\frac{D_i \tau_{ni}}{2} [1 - \exp(\frac{-2h}{\tau_{ni}})]} \quad (21)$$

D_e and D_i are noise diffusion coefficients:

$$D_e = \frac{2\sigma_{ne}^2}{\tau_{ne}} \quad (22)$$

$$D_i = \frac{2\sigma_{ni}^2}{\tau_{ni}} \quad (23)$$

where σ_{ne} and σ_{ni} are standard deviations of the excitatory and inhibitory noise conductances, respectively. All of the parameter values for the noise term can be found in Table 2. The neuron model also handles synaptic conductances, which increase immediately with each spike and then decay exponentially with time constants τ_{synex} for spikes coming from excitatory neurons and τ_{synin} for inhibitory ones. To implement the full neuron model we described it in NESTML (Babu et al., 2021), from which code can be automatically generated for NEST 3.0 (Hahne et al., 2021).

2.1.2. Neuron model variations

We examine the robustness of the model results to simplification of the neuron model described above. The first variation we apply is to disable the intrinsic conductance noise mechanism by setting σ_{ne} and σ_{ni} to 0. This adjustment also allows us to study the susceptibility of the networks to noise in the system. In a second step, we additionally disable the I_{Na} , I_K , and I_M currents, resulting in leaky integrate-and-fire (iaf) neurons. We leave all parameters unrelated to these ion channel currents unchanged, but change the membrane potential threshold V_{thresh} of each population (L2/3-E: -52 mV, L2/3-I: -55 mV, L4-E: -49 mV, L4-I: -55 mV, L5-E: -57.0 mV, L5-I: -65.0 mV) such that the means of the population firing

TABLE 2 Neuronal conductance noise parameters; source definition as in Table 1.

Parameter	Value	Source	Description
τ_{ne}	2.7 ms	Paper	Time constant for the excitatory noise conductance
τ_{ni}	10.5 ms	Paper	Time constant for the inhibitory noise conductance
g_{ne}	12 nS	Paper	Mean conductance of the excitatory noise
g_{ni}	57 nS	Paper	Mean conductance of the inhibitory noise
σ_{ne}	3 nS	Paper	Standard deviation of the excitatory noise conductance
σ_{ni}	6.6 nS	Paper	Standard deviation of the inhibitory noise conductance

rates of the data-based circuit with integrate-and-fire neurons match those of the network with Hodgkin-Huxley neurons as closely as possible (see Supplementary materials for firing rate distributions of all networks).

2.1.3. Synapse model

For the synaptic short-term dynamics, we use the `tsodyks2_synapse` model implemented in NEST. This model implements short-term synaptic plasticity according to Maass and Markram (2002) with the following equations, which are also used in the replicated paper:

$$A_k = w \cdot u_k \cdot R_k \quad (24)$$

where A_k is the amplitude of the postsynaptic potential for the k th spike and w is the synaptic weight. The release probability u_k is given by:

$$u_k = U + u_{k-1}(1 - U) \exp(-\frac{\Delta_{k-1}}{\tau_{fac}}) \quad (25)$$

where U determines the increase in u with each spike, Δ_k denotes the time since the last spike and τ_{fac} is the time constant for recovery from facilitation. R_k is the fraction of synaptic efficacy available for the k th spike and follows:

$$R_k = 1 + (R_{k-1} - u_{k-1}R_{k-1} - 1) \exp(-\frac{\Delta_{k-1}}{\tau_{rec}}) \quad (26)$$

where τ_{rec} is the time constant for recovery from depression. The variables u_k and R_k are initialized with $u_1 = U$ and $R_1 = 1$. The mean values for U , τ_{fac} and τ_{rec} as well as the synaptic delay

TABLE 3 Population type dependent synaptic parameters; source definition as in Table 1.

From/to	Parameter			Source	Description
		E	I		
E	U	0.5 s	0.05 s	Paper	Increase of release probability with each spike
	τ_{rec}	1.1 s	0.125 s	Paper	Time constant for depression
	τ_{fac}	0.05 s	1.2 s	Paper	Time constant for facilitation
	d	1.5 ms	0.8 ms	Code	Synaptic delay
I	U	0.25 s	0.32 s	Paper	Increase of release probability with each spike
	τ_{rec}	0.7 s	0.144 s	Paper	Time constant for depression
	τ_{fac}	0.02 s	0.06 s	Paper	Time constant for facilitation
	d	0.8 ms	0.8 ms	Code	Synaptic delay

depend on the type of their source and target neurons and can be found in Table 3.

These parameters are not fixed for a given ensemble of synapses between a source population j and a target population i ; instead they are drawn from a Gaussian random distribution with a standard deviation of 50% (b_U for U , b_{rec} for τ_{rec} and b_{fac} for τ_{fac}), 10% (b_d for the delay) or 70% (b_w for the weight) of their mean values. As described by Häusler and Maass (2007), all negative values or values bigger than the upper bound of the range (for U) are replaced by values drawn from a uniform distribution between 0 and two times the mean. Note that using a truncated normal distribution leads to a different network activity with higher firing rates (data not shown).

The mean amplitudes A_{ij} of the postsynaptic potentials for the connections between populations j and i , which are needed to calculate their mean weights, can be found in Figure 1. With this we get the value for their weight w by:

$$w_{ex} = \frac{A_{ij}}{|E_{ex} - V_{mean}|} \quad (27)$$

for excitatory synapses and

$$w_{in} = \frac{A_{ij}}{|E_{in} - V_{mean}|} \quad (28)$$

for inhibitory ones. E_{ex} and E_{in} are the excitatory and inhibitory synaptic reversal potentials and V_{mean} is the mean membrane voltage of a neuron without input. All values of the synaptic parameters can be found in Tables 3, 4.

2.1.4. Network models

In this section, we describe the different network models implemented in the original work and in this replication. All circuits comprise 560 of the Hodgkin-Huxley neurons described above, unless otherwise stated. Another common feature shared by six of the seven circuits is that they are connected by synapses with short-term adaptation as described above. The

exception is the data-based model variant with static synapses, which helps us examine the effects of synaptic dynamics on task performance. Figure 2 shows the histograms of degrees (number of incoming and outgoing synapses) for the different circuits; this serves as the first validation of our work, as they are visually indistinguishable from those presented in Figure 7 of Häusler and Maass (2007), which is the best that can be achieved without access to the original data.

Data-based circuit

The data-based model consists of three layers, each divided into an excitatory and an inhibitory population. Figure 1 illustrates the network's connectivity structure; a specification of the parameters can be found in Tables 3, 4. Since the data on which the circuit is based comes from biological systems with a much larger number of incoming connections per neuron, the synaptic weights in the model are scaled up by a factor S_{RW} to obtain a reasonable network activity. In the paper, the value of this scaling factor is given as $60000/N$ (about 107 for $N = 560$), but in the published code this parameter is calculated as $66825/N$ (about 119 for $N = 560$). We use the second value in our implementation, because it gives a network activity closer to the reported one. The distribution of degrees for this connectivity model (and all following models) can be seen in Figure 2.

Amorphous circuit

The amorphous circuit is derived from the data-based circuit by destroying the laminar connectivity structure: for each connection, we replace the source neuron with a random neuron of the same type (excitatory or inhibitory) and also the target neuron with a random neuron of the same type (excitatory or inhibitory), whereby the new randomly selected neurons are not constrained to belong to the same layer as the ones they replace. Multiple connections between the same neuron pair are excluded. This results in a network that shares most global statistics with the data-based model: number of synapses, their pre- and postsynaptic neuron type and the distribution of all

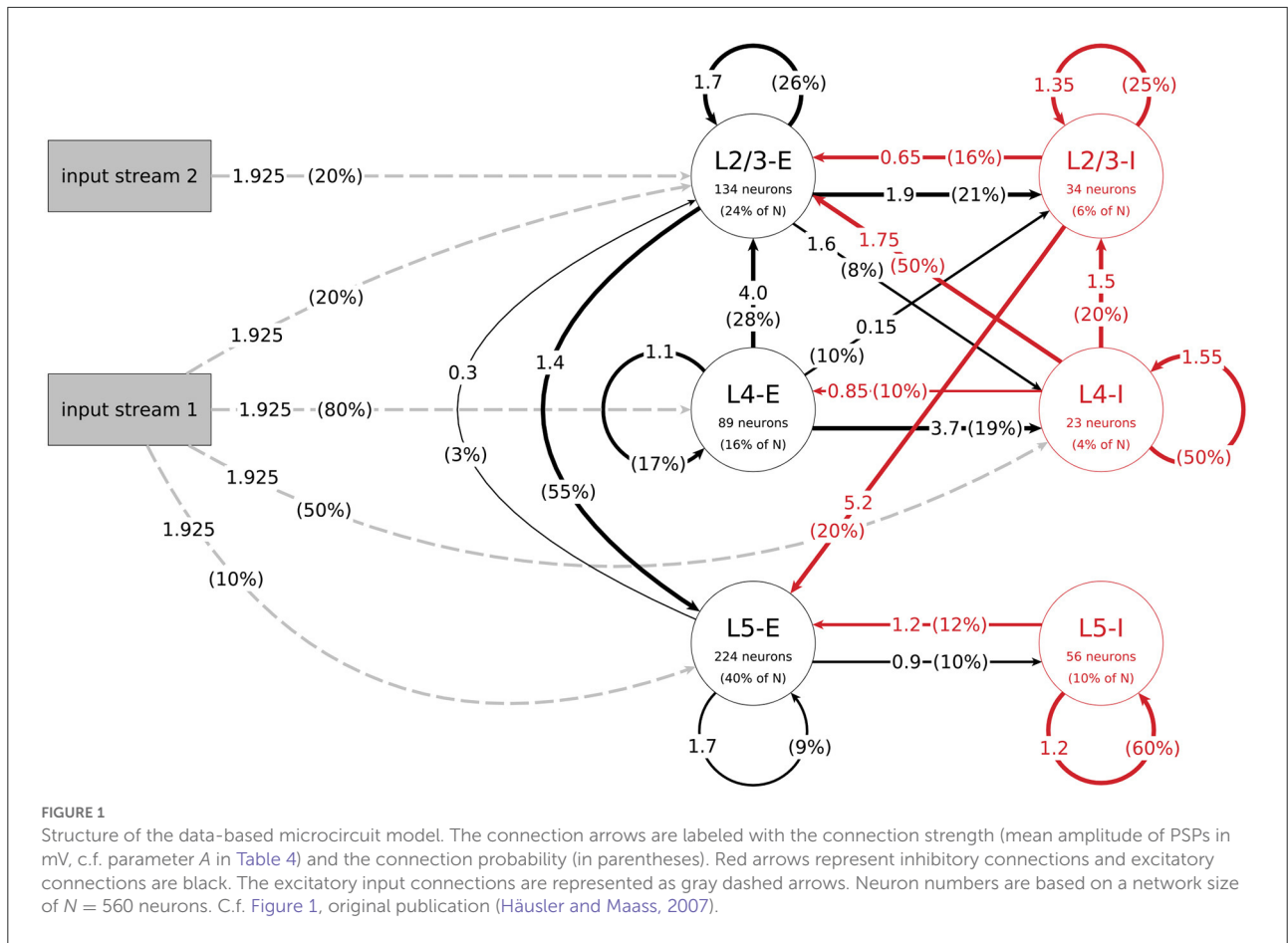
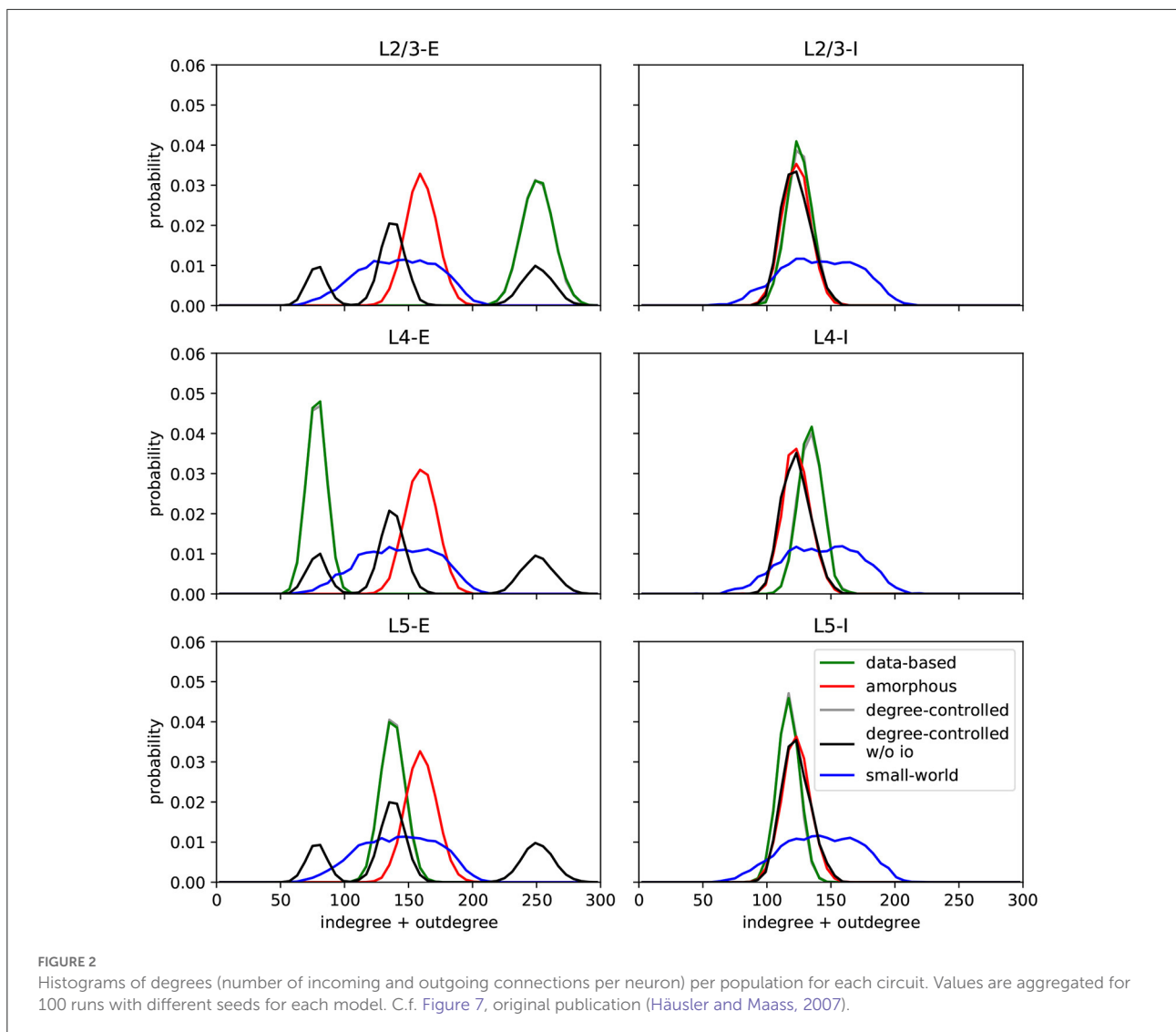


TABLE 4 Synapse parameters.

Parameter	Value	Source	Description
<i>A</i>	see Figure 1	Paper	Mean amplitude of PSPs
<i>A_{input}</i>	1.924779612734342 mV (1.9 mV)	Code (paper)	Mean amplitude of PSPs for input connections
<i>V_{mean}</i>	-65 mV	Destexhe et al., 2001	Mean membrane voltage of a neuron without input
<i>S_{RW}</i>	66,825/ <i>N</i> (60,000/ <i>N</i>)	Code (paper)	Scaling parameter for recurrent connections
<i>S_{RWstatic}</i>	<i>S_{RW}</i> / 73	Experiment	<i>S_{RW}</i> for data-base circuit with static synapses
<i>S₁</i>	14.85 (14)	Code (paper)	Scaling parameter for connections from stream 1
<i>S₂</i>	36.498 (33)	Code (paper)	Scaling parameter for connections from stream 2
<i>w_{ex}</i>	$S_{RW} \cdot \frac{A_{gl}}{ E_{ex} - V_{mean} }$	Code	Maximum conductance for excitatory synapses
<i>w_{in}</i>	$S_{RW} \cdot \frac{A_{gl}}{ E_{in} - V_{mean} }$	Code	Maximum conductance for inhibitory synapses
<i>b_U</i>	0.5	Paper	Factor defining the std. for the distribution of <i>U</i>
<i>b_{rec}</i>	0.5	Paper	Factor defining the std. for the distribution of τ_{rec}
<i>b_{fac}</i>	0.5	Paper	Factor defining the std. for the distribution of τ_{fac}
<i>b_d</i>	0.1	Code	Factor defining the std. for the distribution of <i>d</i>
<i>b_w</i>	0.7	Paper	Factor defining the std. for the distribution of <i>w_{ex}</i> and <i>w_{in}</i>

The source column indicates where the value can be found, searching in the following order: main replicated paper, referenced papers, source code, own experiments. If a value was given in the paper which differs from the one used in the code, the paper value is written in parenthesis.



synaptic parameters such as the weights and the parameters defining the short-term dynamics remain unchanged.

Degree-controlled circuit

The degree-controlled circuit is also derived from the data-based circuit by scrambling its connections. However, in this network, we ensure that the number of incoming and outgoing connections (the degree) for each neuron remains unchanged. To achieve this, we randomly select two synapses whose source neurons are of the same type (excitatory or inhibitory) and whose target neurons are also of the same type, and exchange the target neurons of these synapses. We continue this procedure until none of the original connections remain. Just as in the amorphous circuit, the global statistics of the network are preserved. In addition, the number of incoming and outgoing connections per neuron is the same as in the data-based circuit.

Degree-controlled circuit without input or output specificity

The degree-controlled circuit without input or output specificity is derived from the degree-controlled circuit by changing the neurons to which external input is given and from which the states are read out. We implement this by randomly exchanging the layer assignments of neurons of the same type (excitatory or inhibitory) after recurrently connecting the network, but before connecting the external input streams and readouts.

Small-world network

As introduced by Watts and Strogatz (1998), the small-world network is one in which the underlying undirected graph has small-world properties. Such networks show a higher clustering coefficient than amorphous circuits, while keeping the average shortest path length at a comparable value. Watts and Strogatz

define the local clustering coefficient of a node as the fraction of all possible connections between the node's neighbors that actually exist. It represents how close the neighborhood is to being a clique. The global clustering coefficient of a network is the average of all local clustering coefficients. The shortest path length between two nodes measures the separation of nodes and is defined as the minimum number of links required to get from one node to the other. This shortest path length is averaged over all possible node pairs in the network. We generated a small-world network using the spatial growth algorithm proposed by Kaiser and Hilgetag (2004); first we initialize the network by assigning the position (0.5, 0.5) to a random node, then we perform the following steps:

1. Take a new node and assign it a random position (x, y) with coordinate values drawn from the interval $[0, 1]$.
2. Connect the new node with all other nodes with probabilities defined by:

$$P(u, v) = \beta e^{-\alpha d(u, v)} \quad (29)$$

where $d(u, v)$ is the euclidian distance between the nodes u and v , β is a general density parameter and α is a spatial range parameter, which regulates the dependence of the connection probability on the distance.

3. Repeat steps 1 and 2 until the desired number of nodes has been reached.

By choosing $\alpha = 4$ and $\beta = 1.32$ we obtain small world networks which have a clustering coefficient around 36% and a average shortest path value of about 1.75 links, comparable to those of data-based circuits.

To get the final connections for the network, we randomly assign a direction to every edge and set the weight and other synaptic parameters according to the neurons' population affiliations (see Figure 1 and Table 3). If a neuron pair belongs to two populations which aren't connected in the data-based circuit (see Figure 1), we randomly draw a weight from connection definitions for the same synapse type (excitatory or inhibitory). For example, given a connection from L5-E to L4-I, we would randomly select another excitatory connection such as L4-E to L4-I and use its weight. Since the other synaptic parameters depend only on the type (excitatory or inhibitory) of the connected neurons and not on the exact population affiliation, we can read them out from Table 3 as we did for all other connections.

Data-based circuit with static synapses

This network is identical to the data-based circuit, but with the dynamic synapse model replaced by static synapses. To achieve a similar network activity we have to adjust the scaling parameter S_{RW} . As this value is not explicitly stated by Häusler and Maass (2007), we tuned this parameter by hand to obtain firing rates of the network as close as possible to the data-based circuit, under the condition that no population

is silent. This is achieved at a value $S_{RW}^{st} = S_{RW}/73$ (see Supplementary materials for the resulting firing rate histograms of all networks).

Data-based circuit with random synaptic dynamics

This network is identical to the data-based circuit, except that the short-term dynamics of the data-based network's connections are scrambled. To do this, for each synapse we randomly select one of the four connection types (EE, EI, IE, II) independently of the actual source and target of the synapse. We then draw values for the parameter values U , τ_{rec} , τ_{fac} , and d according to the corresponding distributions for the selected connection type, with mean values as given in Table 3 and standard deviation factors as given in Table 4.

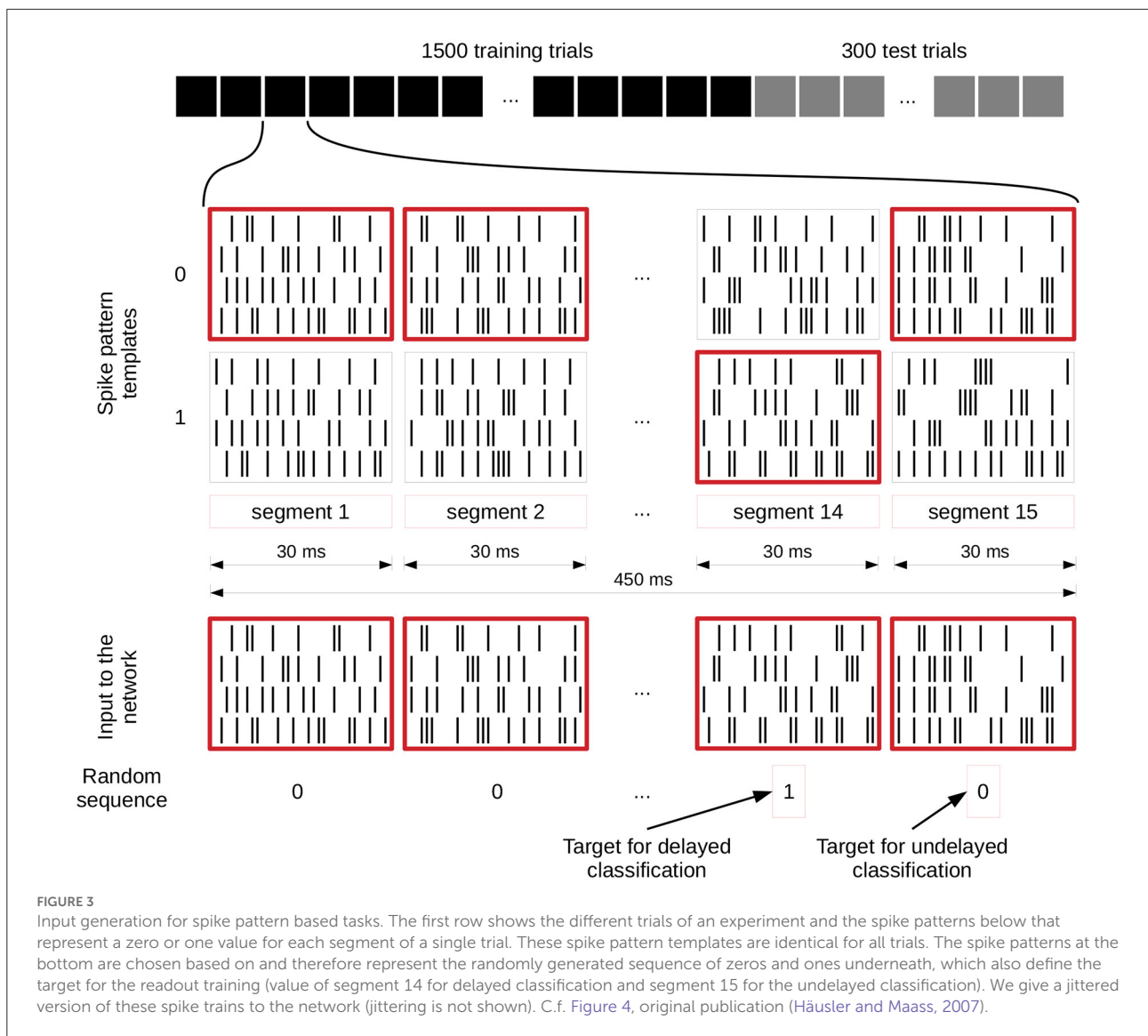
2.2. Tasks

Häusler and Maass (2007) implemented several different tasks to evaluate the computational performance of the different network models. Some of the tasks are based on the classification of precise spatio-temporal spike patterns and for others the circuits need to perform computations on the input firing rates of input spike trains whose spikes are generated by a Poisson process. All tasks are based on inputs given as two input streams which are connected to the network as shown in Figure 1.

Analogously to the scaling of recurrent connections by the factor S_{RW} , the weight values of the synapses from these streams to the circuit are multiplied by their scaling factors S_1 and S_2 . As can be seen in Table 4 the values given in the paper differ from the values in the code; we use the latter in our implementation.

Depending on the task, the input streams consist of either four (rate based tasks) or 40 (spike pattern classification tasks) spike trains. Per trial, 15 segments with a duration of 30 ms are generated, resulting in a spike train of 450 ms for each trial. Only the input for the retroactive spike pattern classification with fixed inter-stimulus input and the more finely resolved memory tasks, which are both further explained in the next section, are exceptions to this scheme.

Figure 3 illustrates how these input streams are generated for the spike pattern based tasks. For each segment, two different spike patterns are generated which encode either a zero or a one and the randomly generated input value (bottom row of zeros and ones in the figure) defines which of them is used in the current trial. These two possible patterns for each segment remain identical for each trial. In this way a sequence of 15 zeros and ones is translated into a set of spike trains of 450 ms length. In addition, each input spike is jittered by a Gaussian distribution with mean 0 ms and a standard deviation of 1 ms. We apply this jittering once per trial to the selected templates and each neuron connected to the input receives the same jittered version of the spike train. Even though some tasks are calculated based on only one of the inputs, both streams are



activated and connected all the time, and the tasks are then evaluated for each input separately.

The tasks are performed by the networks using a reservoir computing approach and thus require readout neurons. We connect two different readouts to the systems: the first readout mimics an excitatory neuron of layer 2/3 and sums up the filtered spike trains of its inputs. Exactly like a normal layer 2/3 neuron, it does not receive input from all possible sources in a linked population, but is randomly connected to a subset of the units based on the corresponding connection probability. The second readout mimics an excitatory layer 4 neuron in the same way. We filter the spikes with an exponential function using a time constant of 15 ms. Note that inhibitory neurons are connected to the spike filtering devices with a negative weight, resulting in negative values. This is important because the readout weights are trained with a linear least squares method

with non-negativity constraints. This results in non-negative readout weights and thus forces the readouts to be in accordance with Dale's principle (Eccles et al., 1954), which states that a neuron releases the same set of transmitters at all of its synapses. The states on which the readouts are trained and tested are the values of the filtered spike trains at the end of each 450 ms trial. A specification of the task parameters can be found in Table 5.

2.2.1. Spike pattern based tasks

We implemented three main spike pattern tasks: spike pattern classification $\text{tcl}_i(t)$, where i denotes the input stream for which the classification is performed, delayed spike pattern classification $\text{tcl}_i(t - \Delta t)$, and the exclusive-or task (XOR). The inputs for all of these tasks are exactly the same; the difference lies in the task-specific training of the readout weights. For the

TABLE 5 Task and training parameters.

Parameter	Value	Source	Description
τ_{filter}	15 ms	Paper	Time constant for spike filtering
T_{train}	1,500	Paper	Default number of training trials
T_{test}	300	Paper	Number of test trials
n_{seg}	15	Paper	Number of segments per trial
t_{seg}	30	Paper	Duration of a single input segment

instantaneous spike pattern classification, the readout weights are trained on the prediction of the value (0 or 1) of the last segment of each trial (segment 15), whereas the target of the delayed classification is the value of the penultimate segment (segment 14), see Figure 3.

In a further set of experiments that go beyond the original study, we use a step duration of 5 ms instead of the standard 30 ms and classify the spike patterns of all 15 segments. As these tasks are all based on only one input, we evaluate them for both input streams separately.

In contrast to this, the XOR task is computed based on the value of the last segment of both input streams. To evaluate the task performances we use a threshold of 0.5 to fix the readout predictions to values of zero or one, and calculate the kappa coefficient between the target output and the predicted output. The kappa coefficient is calculated as:

$$\kappa = \frac{P_0 - P_C}{1 - P_C} \quad (30)$$

where P_0 is the agreement between the target and the observed prediction and P_C is the chance agreement.

In addition to these three task types, we also implemented the retroactive spike pattern classification with fixed inter-stimulus input, which Häusler and Maass (2007) use to evaluate the training convergence in dependence on the number of training trials. The task is to classify spike patterns consisting of four spike trains with a duration of 100 ms after an intervening fixed spike pattern of 100 ms was given to the network in every trial. We implemented this by setting the segment duration to 100 ms, the number of segments per trial to two, the input dimension to four, the second input value of every trial to zero and the first input value as the target for the readout. We also disabled spike jittering for the fixed spike trains between the target stimuli.

2.2.2. Firing rate tasks

In addition to the spike pattern based tasks, computational tasks were also defined using time varying firing rates. The structure of the input streams is similar to the one used in the previously described tasks and the visualization in Figure 3. However, instead of taking one of two pre-generated spike

patterns, we define a target firing rate between 15 and 25 spks/s for each of the 15 segments in both input streams and generate the spike trains based on this rate. The firing rate of the last 15 ms of each trial is used as the target for the readouts. To avoid errors resulting from a division by zero, we ensure that at least one spike is placed in this last 15 ms window of the input streams. The tasks we implemented are the quotient of the two input streams r_1/r_2 and the square of their difference $(r_1 - r_2)^2$. As the kappa coefficient can't be used for these analog prediction tasks, we evaluate the performance on the basis of the Pearson correlation coefficient between the prediction and the target.

2.3. Simulation and analysis framework

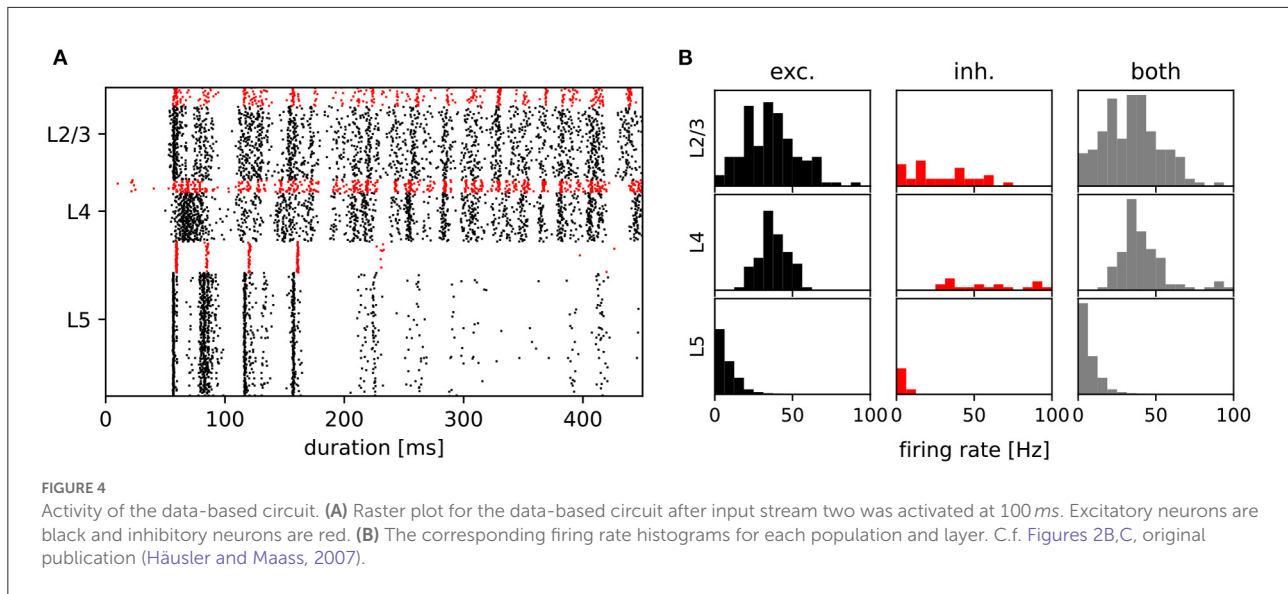
We simulate the spiking neural network experiments with a timestep of 0.2 ms using NEST 3.0 (Hahne et al., 2021). Since the neuron model we describe above is not included in NEST, we implement it using NESTML 4.0.0 (Babu et al., 2021). For all other data analysis and plotting we use Python 3.8.8 and a modified version of the Functional Neural Architectures library (Duarte et al., 2021).

3. Results

3.1. Network activity

After establishing that the degree distributions of the various networks were visually indistinguishable from the published distributions (see Figure 2), we then examined the activity of the data-based network. Figure 4A shows a raster plot for the network with input stream two becoming active at 100 ms, and Figure 4B provides the corresponding firing rate histograms for the six populations and, combined, the three layers (c.f. Figure 1). These plots can be compared with Figures 2B,C of the original publication.

Note that whereas the firing rate histograms in Figure 4B are very similar to those shown in the original paper, the raster plot in Figure 4A exhibits some discrepancies. Most notably, the latency of network activity is longer in our implementation than in the original. Only a few inhibitory layer 4 neurons show earlier activity, and although both figures are based on trials of only 450 ms, this behavior is consistent in our experiments. Less consistent is the measured firing rate of layer 5. In contrast to the original study, which reports a stable firing rate of around 8.5 spks/s in this layer, we observe a range of firing rates between 3 and 9 spks/s for differently seeded runs. A possible explanation for these discrepancies is that in the original code, the values of E_M and V_m are transformed into a different simulation voltage range to compute the non-inactivating K^+ current I_M . For this transformation, values of -70 mV for the resting potential and -40 mV for the threshold potential were used,



rather than the values used in the rest of the study (-80 mV , -30 mV , see Table 1). In our implementation, we elected not to include these transformations in the neuron model, as we could determine neither a biological basis nor a computational advantage for so doing; as shown in the following sections, a qualitative reproduction of the task performances is achieved without such transformations.

3.2. Task performance for the circuit variants

Figure 5 shows the results of the seven main tasks for the data-based and the amorphous circuits. Although the performance values are not identical to the ones in the original study, the values are close and qualitatively reproduce the key finding that the data-based circuit outperforms the amorphous control circuit in every task. The main difference between our results and those reported in the original study is our comparatively low performance at rate based tasks and the delayed spike pattern classification of input stream one (for both circuits).

Additionally, Figure 6A shows the performance of the data-based and amorphous circuit for the retroactive spike pattern classification task with fixed inter-stimulus input for different numbers of training examples. The original study does not specify which input stream was used to generate the corresponding figure in their work (Figure 8); we therefore tested both of them. Our experiments show more similar results for input stream one, and so we use those results as the basis for Figure 6A. As in the original study, the data-based circuit has a lower test and training error than the amorphous circuit

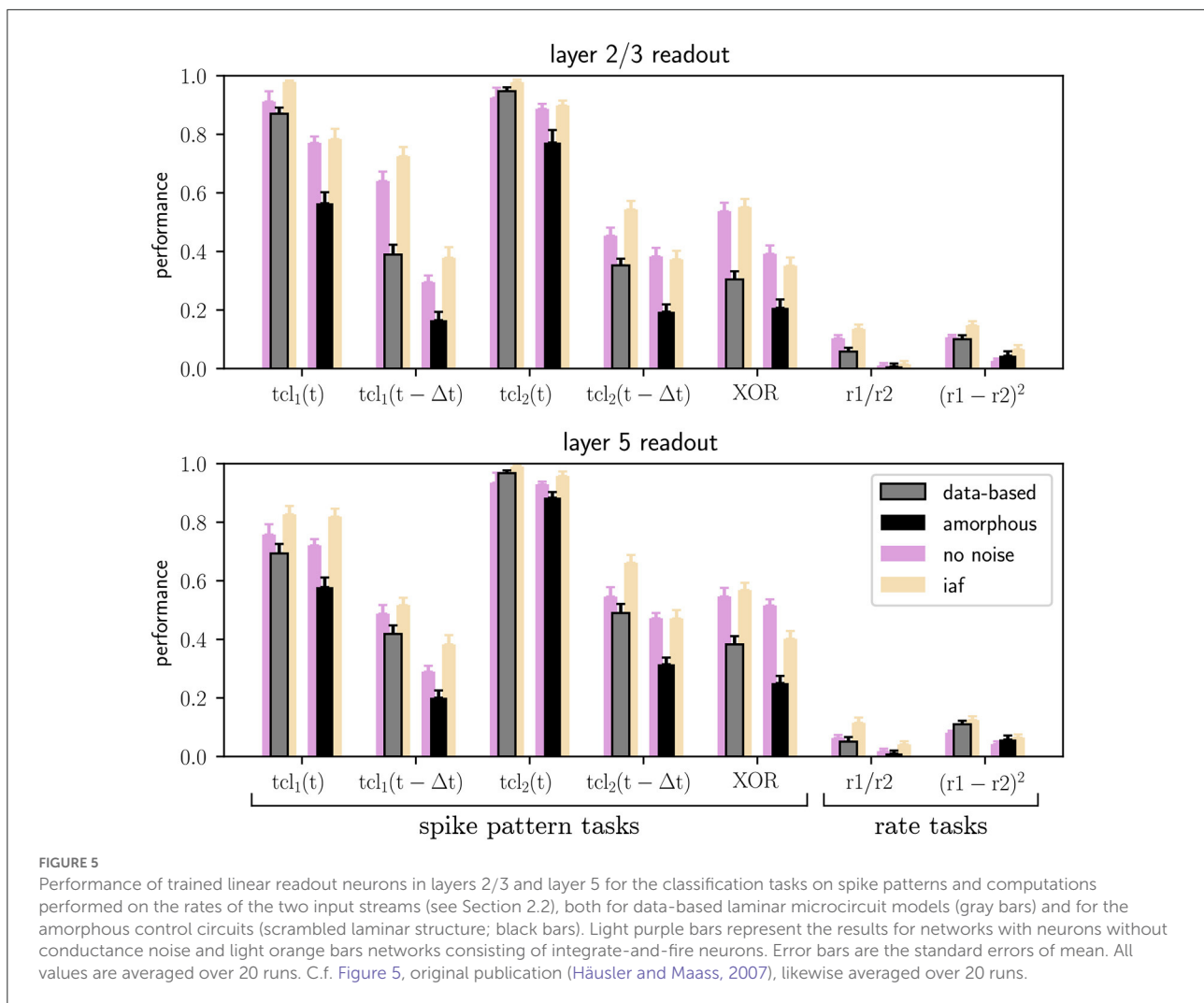
for all sizes of training set. Taken together, Figures 5, 6A support the argument put forward by the original study that a laminar structure has a positive effect on the computational performance of a circuit.

The quantitative performance measures for all circuits (Section 2.1.4) and all tasks (Section 2.2) can be found in Table 6. These results can also be expressed as percentage difference from the performance of the data-based circuit; this analysis is given in Table 7.

The last four rows of both tables show results averaged over both readouts and over a category of tasks. The *memory* row averages over all tasks for which the networks need to memorize earlier inputs [$\text{tcl}_1(t - \Delta t)$ and $\text{tcl}_2(t - \Delta t)$], the *nonlinear* row averages the results over the tasks based on nonlinear computations (XOR, r_1/r_2 and $(r_1 - r_2)^2$) and the *other* row summarizes all other tasks [$\text{tcl}_1(t)$ and $\text{tcl}_2(t)$]. The last row of the tables averages the results over all tasks. In the original paper only the last four rows of Table 7 were presented (Table 2 in Häusler and Maass, 2007).

According to the averaged results, the data-based circuit outperforms all other circuits on all tasks. Here we have broad agreement with the original study, in which only the degree controlled network had slightly superior performance in two task categories.

Examining the disaggregated data in Table 7, we observe that there are 17 instances where a control circuit exhibited superior performance to the data-based circuit. In particular, most networks surpass the data-based circuit on all rate-based tasks. However, as Table 6 shows, the performance for these tasks is very low for all networks, which means that even a small absolute increase in the correlation coefficient results in



a substantial percentage increase. We therefore consider this partial contradiction of the original study to be neuroscientifically uninteresting.

In addition to the rigorous analysis of the effect of circuit connectivity, the original study also considered the influence of network size by increasing the number of neurons within each population of the circuit. Figure 6B shows the dependence of the XOR task performance as a function of the number of neurons in the circuit. As with the original study, our results show a systematically better performance for the data-based circuit over the amorphous circuit, and an increase in performance for both circuits types with increasing circuit size up to 5,000 neurons. While the data-based network still benefits from increasing the network size to 10,000, the performance of the amorphous circuit reaches its maximum value at 5,000 neurons. This effect could not be observed in the original study, as the maximum size of network examined was 1,000 neurons.

3.3. Robustness to neuron model simplifications

The original study demonstrated the computational benefits of lamina-specific connectivity using a fairly complex neuron model. We therefore hypothesize that the details of the neuron model are not relevant to this key finding. To test this hypothesis, we examine the robustness of our dynamical and computational results to variations in the neuron model (see Section 2.1.2). First, we consider the intrinsic noise mechanism. As shown by the raster plots and firing rate histograms in Figure 7, the firing activity in the networks does not change significantly for the data-based and amorphous circuits in the absence of intrinsic noise. Moreover, we observe that the data-based connectivity structure is still superior to all other connectivity patterns in all task types, which Figure 5 (light purple bars) illustrates for the comparison with the amorphous circuit

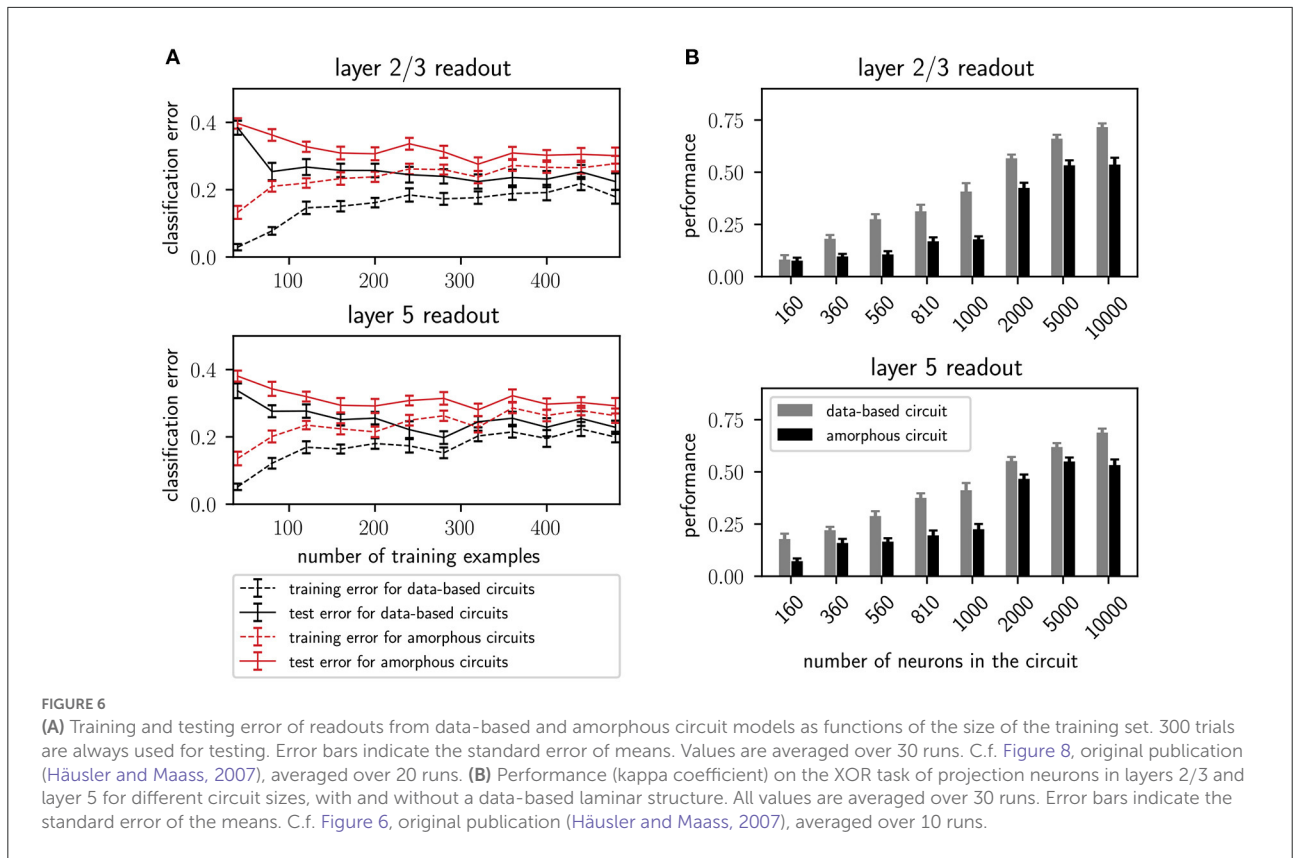


TABLE 6 Performance measures for all networks and all tasks.

Tasks/circuits	Data-based	Amorphous	Small-world	DC	DC (no io)	Random dynamics	Static synapses
tc ₁ (t) (L23)	0.87	-0.305	-0.17	-0.106	-0.235	-0.117	0.013
tc ₂ (t) (L23)	0.947	-0.175	-0.171	-0.097	-0.168	-0.44	-0.067
tc ₁ (t) (L5)	0.693	-0.115	-0.145	-0.106	-0.049	-0.407	-0.069
tc ₂ (t) (L5)	0.968	-0.084	-0.117	-0.065	-0.191	-0.469	-0.159
tc ₁ (t-Δt) (L23)	0.389	-0.224	-0.024	-0.061	-0.131	+0.021	-0.056
tc ₂ (t-Δt) (L23)	0.352	-0.158	-0.02	+0.071	-0.081	-0.185	-0.091
tc ₁ (t-Δt) (L5)	0.418	-0.217	-0.073	-0.117	-0.101	-0.136	-0.258
tc ₂ (t-Δt) (L5)	0.49	-0.175	-0.119	-0.007	-0.187	-0.259	-0.196
XOR (L23)	0.304	-0.097	-0.145	-0.053	-0.114	-0.174	-0.061
r1/r2 (L23)	0.058	-0.05	+0.083	+0.043	+0.076	+0.073	-0.007
(r1 - r2) ² (L23)	0.1	-0.056	+0.026	+0.028	+0.012	+0.006	-0.013
XOR (L5)	0.383	-0.132	-0.206	-0.102	-0.198	-0.206	-0.095
r1/r2 (L5)	0.051	-0.04	+0.069	+0.064	+0.116	+0.029	-0.025
(r1 - r2) ² (L5)	0.11	-0.052	+0.026	+0.017	+0.014	-0.02	-0.067
Memory	0.43	-0.183	-0.07	-0.001	-0.147	-0.17	-0.135
Nonlinear	0.168	-0.072	-0.025	-0.001	-0.016	-0.049	-0.045
Other	0.938	-0.162	-0.144	-0.083	-0.196	-0.373	-0.093
All	0.463	-0.129	-0.072	-0.024	-0.105	-0.176	-0.084

Spike based tasks are evaluated with the kappa coefficient and rate based tasks with the correlation coefficient. The data-based column gives the absolute value and the other columns show the difference from this value. The best performance per task/row is marked in bold. Gray/blue shading denotes tasks from the categories memory/nonlinear. All values are averaged over 20 runs (10 runs in original paper).

TABLE 7 Performance measures for all control networks and all tasks expressed as average difference (in percent) from the performance of the data-based circuit.

Taks/circuits	Amorphous	Small-world	DC	DC (no io)	Random dynamics	Static synapses
tcl ₁ (t) (L23)	-35.1	-19.6	-12.2	-27.0	-13.4	1.5
tcl ₂ (t) (L23)	-18.5	-18.1	-10.2	-17.8	-46.5	-7.1
tcl ₁ (t) (L5)	-16.6	-20.9	-15.3	-7.2	-58.7	-10.0
tcl ₂ (t) (L5)	-8.6	-12.1	-6.6	-19.6	-48.4	-16.4
tcl ₁ (t-Δt) (L23)	-57.5	-6.2	-15.6	-33.8	5.4	-14.4
tcl ₂ (t-Δt) (L23)	-44.9	-5.9	20.1	-23.1	-52.6	-25.9
tcl ₁ (t-Δt) (L5)	-52.0	-17.5	-28.1	-24.3	-32.5	-61.8
tcl ₂ (t-Δt) (L5)	-35.7	-24.3	-1.4	-38.2	-52.9	-40.0
XOR (L23)	-31.9	-47.8	-17.6	-37.5	-57.4	-20.2
r1/r2 (L23)	-86.8	142.6	74.1	130.8	126.5	-11.2
(r1 - r2) ² (L23)	-56.1	26.3	28.2	11.8	6.3	-12.7
XOR (L5)	-34.5	-53.9	-26.7	-51.8	-53.9	-24.8
r1/r2 (L5)	-79.2	136.1	126.4	227.8	57.4	-48.6
(r1 - r2) ² (L5)	-47.1	23.7	15.5	12.3	-18.0	-61.2
Memory	-42.5	-16.4	-0.2	-34.1	-39.7	-31.3
Nonlinear	-42.5	-14.7	-0.3	-9.5	-29.0	-26.6
Other	-17.3	-15.3	-8.8	-20.9	-39.8	-9.9
All	-27.9	-15.5	-5.2	-22.6	-38.1	-18.2

Positive values indicating a performance improvement with respect to the data-based circuit are marked in bold. Gray/blue shading denotes tasks from the categories memory/nonlinear. All values are averaged over 20 runs. C.f. Table 2 in original publication (Häusler and Maass, 2007).

(see [Supplementary materials](#) for the summarized performance measures for all circuit types). [Figure 5](#) also shows that networks without noise (both data-based and amorphous circuits) perform slightly better than their noisy counterparts in most of the tasks performed, and even considerably better in the nonlinear XOR task.

Second, we reduce the neuron model from a Hodgkin-Huxley to a much simpler integrate-and-fire neuron model. To obtain firing rate ranges in the data-based circuit as close as possible to those of the network with Hodgkin-Huxley neurons, we adjust V_{thresh} of the integrate-and-fire neuron model to a different value for each population (see Section 2.1.2 for the parameter values and [Supplementary materials](#) for the firing rate distributions). The top right part of [Figure 7](#) shows the corresponding raster and firing rate plots.

Also among the circuits consisting of integrate-and-fire units, the data-based network has the best task performance (see [Supplementary materials](#) for the summarized performance measures). Moreover, [Figure 5](#) shows similar results for the Hodgkin-Huxley neural networks without noise (light purple bars) and the iaf circuits (light orange bars), with the XOR task values of the amorphous circuits showing the most noticeable difference.

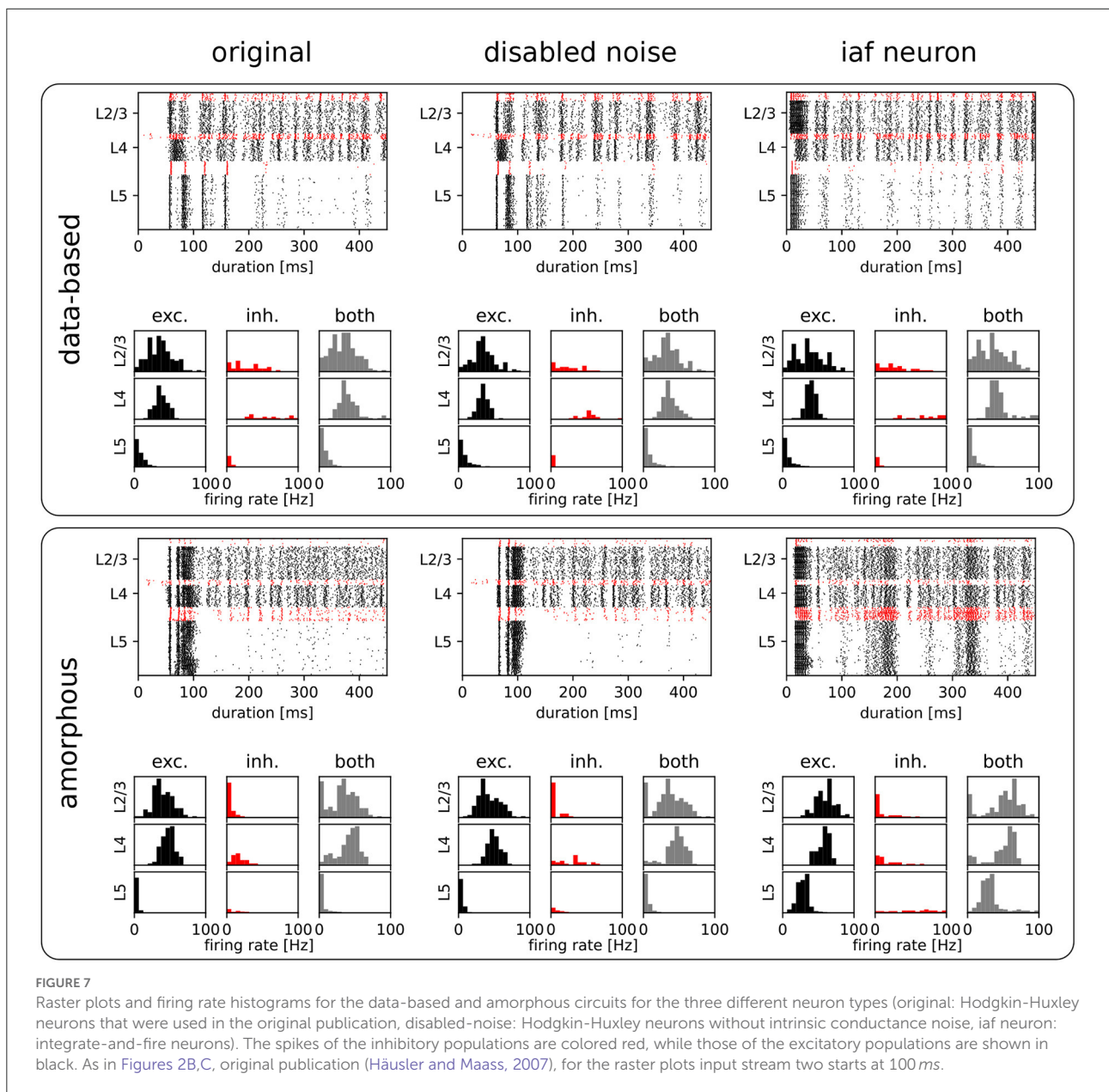
We conclude that these results confirm our hypothesis that the superiority of the data-based connectivity structure does

not depend on the specifics of the neuron model. Moreover, they reveal that a reduction in complexity even leads to an increase in performance on the tasks conducted. We hypothesize that the dynamics of the Hodgkin-Huxley model, which are much more intricate than those of integrate-and-fire neurons, may effectively act as an additional noise source that reduces task performance.

3.4. Detailed memory tasks

To get further insight into the memory capabilities of the networks, we devised a modification of the retroactive spike pattern classification task, namely reducing the step size from 30 ms to 5 ms and classifying the spike patterns of all 15 segments (see Section 2.2). This gives our view on retroactive spike pattern classification a six times higher resolution with one data point for every 5 ms interval instead of only every 30 ms, allowing us to determine the memory profile for each circuit variant. Our results for the more detailed memory tasks are summarized in [Figure 8](#).

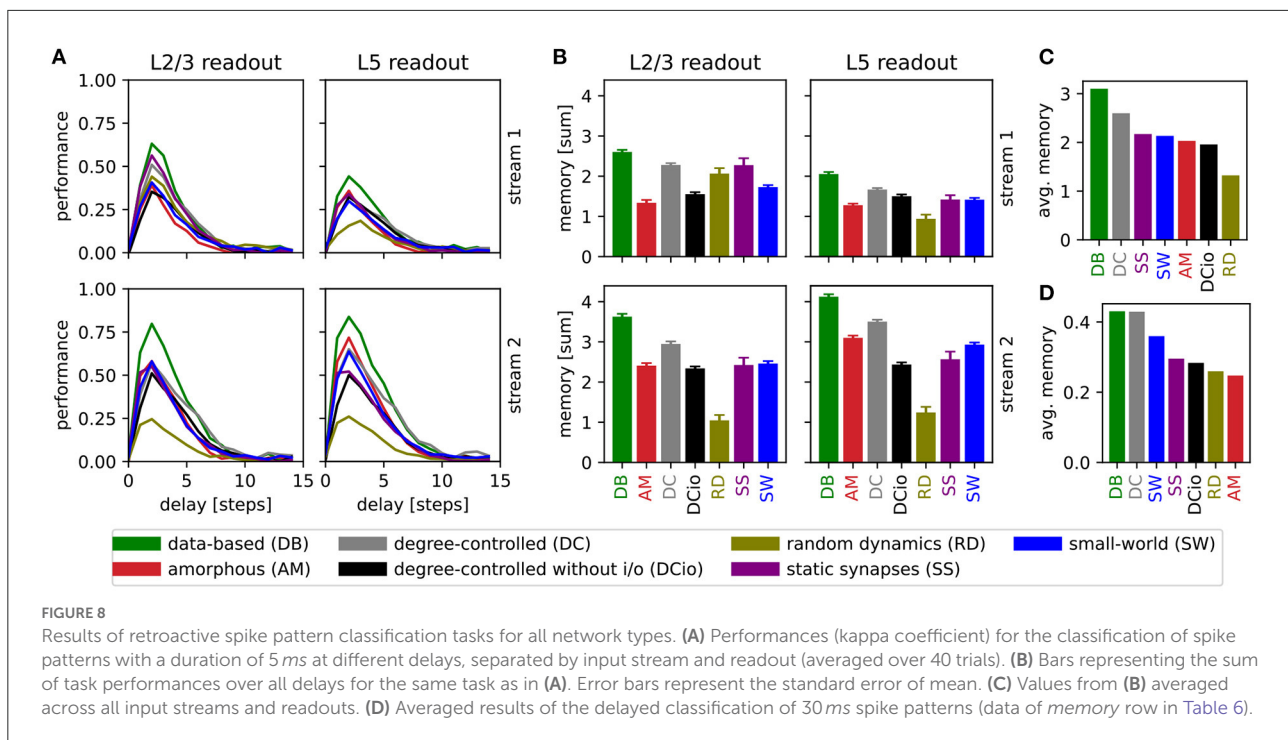
We observe that all network variants require some processing time to reach their peak performance, see [Figure 8A](#). For all combinations of network, input stream, and readout location, the maximum kappa coefficient is reached after a



delay of two steps (10 ms), i.e., the networks have the greatest accuracy in identifying the stimulus inserted two steps before the current one. The only exception is the layer 5 classification of input stream 1 of the data-based network with random synaptic dynamics, which reaches its maximum after a delay of three steps (15 ms). However, in general the performance increases steeply up to delay 2 (10 ms) and then decreases more slowly until all circuits reach a value close to zero at delay 10 (50 ms).

Notably, the performance of the undelayed classification is worse than that of the short-term delayed classification, in contrast to the results presented in Figure 5 for a step of 30 ms. This can be understood by considering that the networks

need more than 5 ms to process the input and generate an informative response from the few neurons on which the readouts are based. One reason for this is synaptic delays, since for example excitatory-to-excitatory synapses already require an average of 1.5 ms to transmit a single spike from a presynaptic to a postsynaptic neuron. With the longer step of 30 ms, the network has plenty of time to respond informatively to the undelayed stimulus, whereas the effects of the previous stimulus have already faded considerably. Likewise, the longer step duration provides greater possibilities for readout weights to be learned that accurately distinguish between stimuli, resulting in a better peak performance for the 30 ms task.



The heights of the bars in Figure 8B indicate the sum of the values for all delays in Figure 8A. This illustrates that not only does the peak performance of the circuit with data-based connectivity surpass that of all other systems at the optimal delay, as shown in the previous line graph, but also that a more general view encompassing the task results for all delays reveals the superiority of this circuit. As the data-based circuit does not retain stimulus information for longer than the other circuits, we conclude that its superior performance must be due to the laminar connectivity enabling it to generate more distinct representations of the input stimuli.

Figure 8C generalizes this view even further by averaging the results shown in Figure 8B over the input stream and readout location, and sorting the networks by performance. Here, the degree-controlled circuit follows the best-performing data-based circuit and the network with random synaptic dynamics has the lowest result. The next four systems (SS, SW, AM, and DCio; Figure 8C) are comparatively at the same level with only minor differences. Although one can not directly compare the averaged sum over multiple delays with the performance based on only a single delay, it is interesting to note that this graph does not display the same order as the results of the previous memory tasks with the longer step of 30 ms shown in Figure 8D and Table 6. While in both cases the data-based circuit is best and the degree-controlled circuit second best, the difference is much smaller for the original memory tasks and the order of the remaining networks is different. For both step durations, the amorphous and the degree-controlled circuit without input and

output specificity show results at similar levels to the network with static synapses. For the 30 ms memory task type the small-world network performs comparatively better than this group, positioning itself in third place, whereas for the detailed memory tasks with 5 ms resolution, the network with random synaptic dynamics performs noticeably worse than the other circuits.

Clearly, the stimulus step duration has an effect on the computational capabilities of the networks and can move a network's peak performance to different delays. Moreover, the ordering differences in Figures 8C,D suggest that the optimal step duration for a network depends on the connectivity structure. However, the good performance of the degree-controlled circuit and especially the data-based circuit for both step durations tested show that both of these systems have a lower dependence on this duration parameter and can more robustly handle stimuli of different lengths.

4. Discussion

4.1. Replicability

Our results demonstrate that we could replicate the circuits of the original study and both confirm and strengthen their key findings. However, we encountered significant challenges during this process and it would have been essentially impossible if we had only had the paper as a source of information. As can be seen in the parameter Tables 1, 3, 4, several of the neuron and

synapse parameters were not given in the paper, or a different value was reported than was used in the code. For example, only by examining the code could we determine the specification of the synaptic delay, namely that it is drawn from a random distribution with a mean value that depends on the connection type (EE, EI, IE, and II).

Likewise, the synaptic time constants were only given in the code and the parameterization of the neuron ion channels showed some discrepancies. In addition to minor differences in the parameters of the Na⁺ ion channel, there was greater variation in the definition of the K⁺ ion channel, which is responsible for the current I_M . Häusler and Maass (2007) provide peak conductance densities for the different ion channels and a single membrane area, which can be used to calculate the peak conductance of every channel. Examining the code reveals that for the standard sodium and potassium ion channels, the reported membrane area is used to calculate the conductivity, however it is based on a different (and unreported) area value for the other potassium ion channel (responsible for I_M). In addition, the conductance density of the latter channel is twice as large as that stated in the paper and different factor r was used in equations 15 and 16 defining the channel dynamics, which is a deviation that is easy to overlook. These differences in parameters significantly alter the activity of the network and led us to disable the M ion channel in early experiments to obtain approximately comparable network responses.

Besides the difficulty in specifying basic synaptic and neuronal parameters, the scaling parameters of the synaptic weights S_{RW} , S_1 and S_2 also caused some problems. All of them were different from the values reported in the paper and it was hard to track them down in the MATLAB code, because the final scaling parameter values were not set directly, but defined by somewhat convoluted calculations distributed over multiple source code files. This was a particularly challenging example of a general problem: as the code was not executable due to its age, it was not possible to simply output the final values of variables, or examine the parameters and dynamic variables of the neurons and synapses. Instead, calculations had to be painstakingly reconstructed by analyzing the source code and replicating the logic.

As a final example of the nature of the replication challenges, the input scaling parameters in the code are approximately—but not exactly—1,000 times smaller than given in the paper, because the input weight to be scaled is approximately—but not exactly—1,000 times bigger than reported. In contrast to the weights inside the network, which are defined in the code as amplitudes of post-synaptic potentials in the same way they are given in the paper, the input weights are defined as a post-synaptic current of 30 nA. This results in the following PSP-amplitude:

$$PSP_{\text{input}} = \frac{PSC_{\text{input}}}{g_L} = 1.924779612734342V \quad (31)$$

instead of the reported 1.9 mV. However, the combination of these differences results in a scaled input weight with the same order of magnitude as the one reported in the paper.

These hurdles to replicating the paper provide a good demonstration of the argument presented in Pauli et al. (2018): whereas provision of source code is the absolute minimum requirement for replicating a study in computational neuroscience, the process is rendered much simpler if appropriate care is taken with the code implementation, e.g., writing modular, encapsulated, well-commented code with separation of parameters and program logic. Moreover, many of the hurdles we encountered would have been substantially reduced if the code had been executable. To foster reproducibility, we therefore recommend (again following Pauli et al., 2018) that models should not be expressed in homebrewed code, as this is unlikely to be maintained. Instead, developing the model using a simulator that is actively developed by a community reduces the maintenance load and increases the likelihood that the model will remain accessible, executable, and part of the scientific discourse for years to come.

5. Conclusion

We analyzed how the lamina structure of a cortical column model affects the computational capabilities of spiking neural networks. In a first step, we replicated the models and experiments described by Häusler and Maass (2007). Although we did not get identical network activity and task results, the activity results and degree histograms are close enough to demonstrate that the replication was successful. Our findings on the tasks defined in the original study confirm their key result, that the degree distribution exhibited by laminar structure of the data-based circuit confers a computational advantage over circuits with modified connectivity patterns that destroy the laminar connectivity whilst maintaining the global statistics of the network. We reach this conclusion by training readout weights to solve tasks based on spike patterns and firing rates that require linear and non-linear computations on two separate input signals and the memorization of prior information.

The microcircuit model at the heart of the original study shares many properties with biological microcircuits. In addition to its data-based structure, it consists of Hodgkin-Huxley neurons with different ion channel dynamics and a conductance-based background noise mechanism, and its synapses exhibit short-term plasticity. For further biological plausibility, its readouts receive only inputs restricted to layer 2/3 and layer 5 specific connections. The readout weights observe Dale's principle (Eccles et al., 1954): excitatory neurons contribute only positive values to the activity function of the readout neuron, and inhibitory neurons only negative values.

Given the superior performance of the data-based circuit, we formulated the hypothesis that the results should be robust with respect to the specifics of the neuron model. We extended the analysis of the original study by decreasing the complexity of the neuron model, first removing the intrinsic noise and then reducing the dynamics to that of an integrate-and-fire neuron. The results confirmed our hypothesis that neuron model details were not important to the key result: in both cases the data-based circuit continued to exhibit superior performance over all other variants. Our results also rule out the possibility that a complex neuron model is necessary for the data-based circuit to reach a good performance, since the two simpler neuron types tested outperform it in the majority of tasks. Likewise, we found that relaxing the biologically motivated restriction on the readout weights increases performance (data not shown).

To obtain a higher temporal resolution in the examination of the memory capabilities of the circuit variants, we additionally extended the original analysis to include retrospective classification of spike patterns of much shorter segments with a duration of 5 ms instead of 30 ms, and classifying all of the segments rather than just the last two. Here we observe a stereotypical memory profile for all circuit types, where the data-based circuit beats the other networks in peak performance and summed reconstruction capability across all delays, with comparable performance for the degree-controlled circuit. However, there is no significant difference between the various networks when comparing the maximum delay up to which the signal can still be at least partially reconstructed. Thus, we conclude that the advantage of the laminar connectivity structure lies primarily in the clarity of the internal representation rather than in significantly longer information retention. These results also highlight the characteristic time scale of the input as a relevant parameter for determining the computational capacities of a spiking neural network.

In future work, we will use our NEST implementation of the data-based microcircuit model, which is now freely available to all researchers, to lay the groundwork for further experiments to investigate the computational properties of cortical columns and to make quantitative comparisons with alternative microcircuit models. In this context it would also be reasonable to tune the network to obtain biologically more realistic long-tailed firing rate distributions with a mean below 1 spks/s instead of the comparatively high activity currently exhibited by the model (about 40 spks/s for layers 2/3 and 4). For the simplified network with integrate-and-fire neurons, this can probably be achieved by adjusting the firing threshold per population based on *in-vivo* data rather than using the activity of the original model as a basis. Similarly, for the network with Hodgkin-Huxley neurons, it is likely that population-level

tuning of neuron parameters and probably adjusted scaling of recurrent weights will be required to achieve the intended firing rates.

From here it is also possible to add other biological details such as additional or different plasticity mechanisms, or to investigate the computational capacities of larger networks using this microcircuit as a basic building block for systems representing the meso- or macroscopic level. This could be achieved, for example, by adjusting the weight scaling parameters along with the network size and connecting multiple differently parameterized instances of the microcircuit using inter-area connectivity that is based on experimental findings.

Data availability statement

The original contributions presented in the study are included in the article/[Supplementary material](#), further inquiries can be directed to the corresponding author.

Author contributions

TS, RD, and AM designed the study and performed the analysis. TS re-implemented the model, performed the simulations, and created the figures. All authors co-wrote the manuscript. All authors contributed to the article and approved the submitted version.

Funding

This project was funded by the Helmholtz Association Initiative and Networking Fund under project number SO-092 (Advanced Computing Architectures, ACA), the Excellence Initiative of the German Federal and State Governments (ERS SDS005), and the Joint Lab Supercomputing and Modeling for the Human Brain. Open access publication funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)-491111487.

Acknowledgments

The authors gratefully acknowledge the computing time granted through JARA on the supercomputer JURECA ([Jülich Supercomputing Centre, 2018](#)) at Forschungszentrum Jülich. We would also like to thank Charl Linssen of the Simulation and Data Laboratory Neuroscience for his reliable and uncomplicated support in the use and application of NESTML and Barna Zajzon for helpful discussions about research related to this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated

organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fnint.2022.923468/full#supplementary-material>

References

- Babu, P. N., Linssen, C., Eppler, J. M., Schulte to Brinke, T., Ziaemehr, A., Fardet, T., et al. (2021). NESTML 4.0. *Zenodo*. doi: 10.5281/zenodo.4740083
- Benureau, F. C. Y., and Rougier, N. P. (2018). Re-run, repeat, reproduce, reuse, replicate: transforming code into scientific contributions. *Front. Neuroinform.* 11, 69. doi: 10.3389/fninf.2017.00069
- Billeh, Y. N., Cai, B., Gratiy, S. L., Dai, K., Iyer, R., Gouwens, N. W., et al. (2020). Systematic integration of structural and functional data into multi-scale models of mouse primary visual cortex. *Neuron* 106, 388.e18–403.e18. doi: 10.1016/j.neuron.2020.01.040
- DeFelipe, J. (2012). The neocortical column. *Front. Neuroanat.* 6, 22. doi: 10.3389/fnana.2012.00022
- Destexhe, A., and Paré, D. (1999). Impact of network activity on the integrative properties of neocortical pyramidal neurons *in vivo*. *J. Neurophysiol.* 81, 1531–1547. doi: 10.1152/jn.1999.81.4.1531
- Destexhe, A., Rudolph, M., Fellous, J.-M., and Sejnowski, T. (2001). Fluctuating synaptic conductances recreate *in vivo*-like activity in neocortical neurons. *Neuroscience* 107, 13–24. doi: 10.1016/S0306-4522(01)00344-X
- Duarte, R., Zajzon, B., Schulte to Brinke, T., and Morrison, A. (2021). Functional neural architectures. *Zenodo*. doi: 10.5281/zenodo.5752597
- Eccles, J. C., Fatt, P., and Koketsu, K. (1954). Cholinergic and inhibitory synapses in a pathway from motor-axon collaterals to motoneurons. *J. Physiol.* 126, 524–562. doi: 10.1113/jphysiol.1954.sp005226
- Habenschuss, S., Jonke, Z., and Maass, W. (2009). Stochastic computations in cortical microcircuit models. *PLoS Comput. Biol.* 9, e1003311. doi: 10.1371/journal.pcbi.1003311
- Hahne, J., Diaz, S., Patronis, A., Schenck, W., Peyser, A., Graber, S., et al. (2021). NEST 3.0.
- Harris, K. D., and Shepherd, G. M. (2015). The neocortical circuit: themes and variations. *Nat. Neurosci.* 18, 170–181. doi: 10.1038/nn.3917
- Häusler, S., and Maass, W. (2007). A statistical analysis of information-processing properties of lamina-specific cortical microcircuit models. *Cereb. Cortex* 17, 149–162. doi: 10.1093/cercor/bhj132
- Häusler, S., Schuch, K., and Maass, W. (2009). Motif distribution, dynamical properties, and computational performance of two data-based cortical microcircuit templates. *J. Physiol. Paris* 103, 73–87. doi: 10.1016/j.jphysparis.2009.05.006
- Horton, J. C., and Adams, D. L. (2005). The cortical column: a structure without a function. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 360, 837–862. doi: 10.1098/rstb.2005.1623
- Jülich Supercomputing Centre (2018). JURECA: modular supercomputer at Jülich supercomputing centre. *J. Large-scale Res. Facilit.* 4, A132. doi: 10.17815/jlsrf-4-121-1
- Kaiser, M., and Hilgetag, C. C. (2004). Spatial growth of real-world networks. *Phys. Rev. E* 69, 036103. doi: 10.1103/PhysRevE.69.036103
- Maass, W. (2014). Noise as a resource for computation and learning in networks of spiking neurons. *Proc. IEEE* 102, 860–880. doi: 10.1109/JPROC.2014.2310593
- Maass, W., and Markram, H. (2002). Synapses as dynamic memory buffers. *Neural Networks* 15, 155–161. doi: 10.1016/S0893-6080(01)00144-7
- Mainen, Z. F., Huguenard, J. R., and Sejnowski, T. J. (1995). A model of spike initiation in neocortical pyramidal neurons. *Neuron* 15, 1427–1439. doi: 10.1016/0896-6273(95)90020-9
- Markram, H., Müller, E., Ramaswamy, S., Reimann, M. W., Abdellah, M., Sanchez, C. A., et al. (2015). Reconstruction and simulation of neocortical microcircuitry. *Cell* 163, 456–492. doi: 10.1016/j.cell.2015.09.029
- McDonnell, M. D., and Ward, L. M. (2011). The benefits of noise in neural systems: bridging theory and experiment. *Nat. Rev. Neurosci.* 12, 415–425. doi: 10.1038/nrn3061
- McDougal, R. A., Morse, T. M., Carnevale, T., Marengo, L., Wang, R., Migliore, M., et al. (2017). Twenty years of ModelDB and beyond: building essential modeling tools for the future of neuroscience. *J. Comput. Neurosci.* 42, 1–10. doi: 10.1007/s10827-016-0623-7
- Mountcastle, V. B. (1997). The columnar organization of the neocortex. *Brain* 120, 701–722. doi: 10.1093/brain/120.4.701
- Pauli, R., Weidel, P., Kunkel, S., and Morrison, A. (2018). Reproducing polychronization: a guide to maximizing the reproducibility of spiking network models. *Front. Neuroinform.* 12, 46. doi: 10.3389/fninf.2018.00046
- Plesser, H. E. (2018). Reproducibility vs. replicability: a brief history of a confused terminology. *Front. Neuroinform.* 11, 76. doi: 10.3389/fninf.2017.00076
- Potjans, T. C., and Diesmann, M. (2014). The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model. *Cereb. Cortex* 24, 785–806. doi: 10.1093/cercor/bhs358
- Rasch, M. J., Schuch, K., Logothetis, N. K., and Maass, W. (2011). Statistical comparison of spike responses to natural stimuli in monkey area V1 with simulated responses of a detailed laminar network model for a patch of V1. *J. Neurophysiol.* 105, 757–778. doi: 10.1152/jn.00845.2009
- Schulte to Brinke, T., Duarte, R., and Morrison, A. (2022). Source code for “Characteristic columnar connectivity caters to cortical computation: replication, simulation and evaluation of a microcircuit model”. *Zenodo*. doi: 10.5281/zenodo.7037649
- Thomson, A. M. (2002). Synaptic connections and small circuits involving excitatory and inhibitory neurons in layers 2–5 of adult rat and cat neocortex: triple intracellular recordings and biocytin labelling *in vitro*. *Cereb. Cortex* 12, 936–953. doi: 10.1093/cercor/12.9.936
- Watts, D. J., and Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *Nature* 393, 440–442. doi: 10.1038/30918
- Wiesenfeld, K., and Moss, F. (1995). Stochastic resonance and the benefits of noise: from ice ages to crayfish and SQUIDS. *Nature* 373, 33–36. doi: 10.1038/373033a0