

Phenomenological modeling of diverse and heterogeneous synaptic dynamics at natural density

Agnes Korcsak-Gorzo^{†, ‡, 1, 2}, Charl Linssen^{†, ‡, 3, 1, 8}, Jasper Albers^{1, 2},
Stefan Dasbach^{1, 4, 16}, Renato Duarte^{5, 1, 14, 15}, Susanne Kunkel^{6, 7},
Abigail Morrison^{1, 8}, Johanna Senk^{1, 12}, Jonas Stapmanns^{1, 2, 13},
Tom Tetzlaff¹, Markus Diesmann^{1, 9, 2, 10}, Sacha J. van Albada^{1, 11}

June 22, 2024

[†] equal contribution

[‡] corresponding authors: a.korcsak-gorzo@fz-juelich.de (AKG), c.linssen@fz-juelich.de (CL)

1. Institute for Advanced Simulation (IAS-6), Jülich Research Centre, Jülich, Germany
2. Department of Physics, Faculty 1, RWTH Aachen University, Aachen, Germany
3. Simulation and Data Lab (SDL) Neuroscience, Jülich Supercomputing Centre (JSC), Jülich Research Centre, Jülich, Germany
4. Institute of Energy and Climate Research - Plasma Physics (IEK-4), Jülich Research Centre, Jülich, Germany
5. Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen, Netherlands
6. Neuromorphic Software Ecosystems (PGI-15), Jülich Research Centre, Jülich, Germany
7. Faculty of Science and Technology, Norwegian University of Life Sciences, Ås, Norway
8. Department of Computer Science 3 - Software Engineering, RWTH Aachen University, Aachen, Germany
9. JARA-Institute Brain Structure-Function Relationships (INM-10), Jülich Research Centre, Jülich, Germany
10. Department of Psychiatry, Psychotherapy, and Psychosomatics, Medical School, RWTH Aachen University, Aachen, Germany
11. Institute of Zoology, University of Cologne, Cologne, Germany
12. Sussex AI, School of Engineering and Informatics, University of Sussex, Brighton, UK
13. Department of Physiology, University of Bern, Germany
14. Center for Neuroscience and Cell Biology (CNC), University of Coimbra, Coimbra, Portugal
15. Centre for Innovative Biomedicine and Biotechnology (CIBB), University of Coimbra, Coimbra, Portugal
16. Heinrich Heine University Düsseldorf, Faculty of Mathematics and Natural Sciences, Düsseldorf, Germany

Abstract

This chapter sheds light on the synaptic organization of the brain from the perspective of computational neuroscience. It provides an introductory overview on how to account for empirical data in mathematical models, implement such models in software, and perform simulations reflecting experiments. This path is demonstrated with respect to four key aspects of synaptic signaling: the connectivity of brain networks, synaptic transmission, synaptic plasticity, and the heterogeneity across synapses. Each step and aspect of the modeling and simulation workflow comes with its own challenges and pitfalls, which are highlighted and addressed.

Keywords

simulation, modeling, synaptic organization, spiking neural networks, plasticity, heterogeneity, phenomenological model, connectivity

1 Introduction

Creating mathematical models from experimental neurophysiological data has grown into an established and essential method for investigating the brain. Based on these mathematical models and exploiting the upswing of affordable and powerful computing architectures over the last few decades, a new sub-field concerned with the computational modeling of neurobiological systems has emerged. The discipline using mathematical modeling and analysis methods to understand principles of brain organization, dynamics, and function is called *computational neuroscience*. This discipline is also sometimes referred to as theoretical or mathematical neuroscience, each term having its own slightly different emphasis. One of the most challenging subjects of this comparatively young domain is the synaptic organization of the brain. This chapter reviews the status quo of synaptic modeling approaches. It is targeted primarily at experimentalists and aims to provide insight into ways data can be used to build mathematical or computational models.

The methods described in the previous chapters of this volume reveal diverse dynamical processes and heterogeneous components involved in synaptic signaling at various spatial and temporal scales. This variability is amplified by the size and complexity of neurobiological systems: both the density and the total number of synapses in mammalian brains are impressive, the former being on the order of 10^9 per cubic millimeter in the cerebral cortex (1) and the latter being estimated as

roughly 5×10^{14} in the human brain (2). Each cubic millimeter of the human brain contains on the order of $10^4 - 10^5$ neurons adding up to about 10^{11} neurons in the brain as a whole (3).

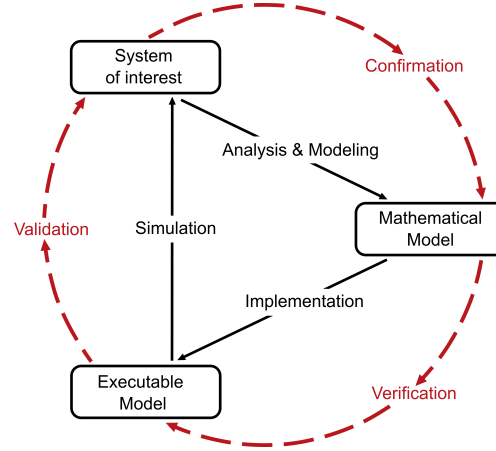


Figure 1: **Cycle of modeling and simulation.** The empirical data from the brain structure under study (the “system of interest”) is first mathematically modeled and then implemented in software. Closing the loop, simulation results can be compared to experimental recordings. Reproduced with permission from Fig. 1 in Trensch et al. (4).

How can we model such a heterogeneous, complex, and dense large-scale system? The process of modeling and simulation can be understood as a cycle, as depicted in Fig. 1. First, the experimental results recorded from the system of interest (here, the synaptic organization of the brain) are analyzed, and a mathematical model is formulated. Then, the mathematical model is translated into computer language, i.e., into an executable model that implements the mathematical operations needed to simulate the model. Finally, the model of the system is executed, whereby this simulation is the numerical analog to an experiment. This process yields results that can be compared with the experimental results. In turn, this comparison may deliver outcomes that can be used to improve the mathematical and computational models. Comparisons between the system of interest, the mathematical model, and the executable model ensure quality control. In general, three types of checks for correctness can be distinguished (4): *Confirmation* ensures that the mathematical formulation applies to the system of interest, *verification* that the executable model sufficiently represents the mathematical model, and *validation* that the simulation outcome is consistent with and predictive of the system of interest. This chapter focuses on the inner triangle of arrows: the practical methods to formulate mathematical models in an informed way, to translate them into manageable and correct executable models, to run simulations, and to inform further modeling choices using

the obtained data.

The challenge of computational neuroscience is to analyze the rich dynamics of neuronal systems and abstract their complexity into mathematical models that still capture essential characteristics of the experimental findings. At the same time, these models should be simple enough to be tractable and generalizable and thus reveal possible laws that govern the dynamical system. A fundamental question in this endeavor is what processes and variables are of interest and best describe the data. In synaptic organization, candidates include the connection strengths, synaptic time constants, delays, vesicle release characteristics, synaptic plasticity, and neuromodulation.

Developing this thought further, a modeler needs to decide on the scale the model represents and how to parameterize it. It is advisable to constrain the number of model parameters to a minimal set that answers a specific research question. Limiting the parameter space increases the tractability, mechanistic interpretability, and robustness of the model and reduces the risk of overfitting. However, capturing biological detail and enhancing the direct link between parameters and their biological counterparts can usually only be done with a large set of parameters. Heterogeneity can be represented by introducing parameter value distributions, leading to additional parameters characterizing the dispersion and possibly higher-order properties of the corresponding distributions. Overall, a suitable parameterization involves a tradeoff between the model's controllability and biological plausibility.

These decisions on which aspects of the system to express as variables and the choice of the corresponding model equations are abstraction steps: they formalize a hypothesis on which features are germane to the question at hand and which mathematical descriptions are appropriate for capturing the phenomena of interest (see Section 7.8). In general, this abstraction can be approached from two different directions. The *bottom-up* approach starts from the low-level properties of the neurons and synapses making up the system and models the complexity step by step in the hope of achieving realistic dynamical and functional properties. However, one major point of modeling is to improve our understanding of a system. Given that the starting point is a poor understanding, this approach suffers from the fundamental problem that essential features might be abstracted away or obfuscated by an abundance of less relevant details. Another point can be to provide accurate predictions, even if we do not understand the model. The opposite approach, *top-down* modeling, starts from the high-level dynamical, functional, or behavioral properties one would like to capture and then proposes concrete implementations. The drawback of this approach is that a model created in this way might not fully conform to biology, so

it is difficult to draw conclusions about the brain. One solution is to use different degrees of abstraction at different scales to arrive at an understanding of the system, which is the motivation behind *multi-scale modeling*. Ideally, biological realism is incrementally enhanced through cycles of data comparison and refinement (see Fig. 1 and Section 7.1).

Some neurophysiological observations can be modeled with analytically solvable equations, i.e., in an exact way and usually with pen and paper. However, various simplifying assumptions generally flow into such abstractions, and deriving an analytical solution to a model's equations becomes less feasible as its complexity increases. For such cases, numerical solutions can provide a useful alternative. This computational approach tends to be slower, but it can validate the analytical approach by requiring fewer simplifying assumptions and it may even provide novel theoretical insight.

Simple small network models frequently consist of equations that can be solved analytically or calculated numerically with few computational resources. However, both numerical and analytical approaches reach certain limits when attempting to replicate realistic neuron numbers in the volume of the brain region under consideration. As the number of neurons increases, the number of connections grows quadratically in networks without spatial dependence and linearly for distant neurons in models incorporating spatial dependence since most connections are local. To approximate *natural density*, analytical techniques like mean-field theory sacrifice biological specificity. With sufficient computing power, numerical methods may solve model equations at natural density. A limitation is that, to date, this is only possible for small brains or small portions of larger brains.

We restrict the scope of this chapter to phenomenological models, which represent the empirical relationship between phenomena without explaining the reason for the interaction. We neglect the molecular level or ultrastructure, i.e., structures visible at magnifications higher than that provided by standard optical light microscopy. Furthermore, we address spiking neuron models, mainly so-called point or few-compartment neurons, which neglect the precise morphology of the neuron, as the effective dynamics of a morphologically complex neuron can often already be meaningfully captured by such models.

The models are usually translated to be executable by a computer, i.e., implemented in one of the various computer languages. One or an interconnected set of dynamical model components representing neurobiological entities such as neurons or synapses is *simulated*, i.e., evolved in time, for a specific duration with a set of parameters, initial conditions, and stimuli.

The generic framework that can numerically solve the dynamical equations of various models is called a *simulator*. It solves complex interactions with many coupled differential equations typically incrementally in discrete time steps. Spiking neuronal network simulators also communicate spike events from senders to receivers. The event-based nature of synaptic interactions in the form of action potentials is advantageous for the efficiency of the simulation. Furthermore, a simulator can record dynamical state variables in the network or other observables like connectivity and apply stimuli during simulation.

Since the simulator needs to organize and maintain the appropriate data structures in computer memory, a substantial amount of RAM may also be required, depending on the simulated neurobiological system. Biologically realistic network models are often large-scale to reflect or approach the natural density of neurons and synapses, have additional computational overhead due to heterogeneity, and are typically simulated for a long time, e.g., to gather statistics or study behavioral timescales. Thus, a simulator should be efficient and scale to high-performance architectures in terms of processing and memory usage.

In addition to these performance aspects, criteria for a good simulator include functional completeness, numerical accuracy, and reproducibility of results. Furthermore, a simulator increases its value for the community if the available models are relevant to many members and the documentation is comprehensive and easy to understand. Developing a simulator that fulfills all these criteria and supports diverse models is complex and time-intensive. Consequently, simulators with peer-reviewed collections of implemented models are continuously developed as a community effort and shared as software packages to benefit the field of computational neuroscience.

From the range of existing simulators for biologically inspired neuronal networks, this chapter focuses on NEST (5), an open-source software tool designed to simulate anywhere from small to large-scale networks of diverse spiking neuron models, and its associated domain-specific modeling language NESTML (6) that facilitates the creation of new neuron and synapse models. Other simulators with various scientific foci and special areas of application include NEURON (7), Brian (8), Nengo (9), Arbor (10), and ANNarchy (11). Some common (simulator-agnostic) interfaces are provided by PyNN (12) and the modeling language NeuroML (13).

This chapter is structured as follows. Each section from Section 2 to Section 6 presents a two-step recipe to go from experimental data on a specific feature or mechanism of the synaptic organization to simulations: first, how to mathematically model experimental data, and second, how to simulate this model. Section 2 starts with the most simplified view of brain circuitry, namely binary connections

between neurons. This view is advanced in Section 3 to the notion of weighted connections. Then, dynamics is introduced to the existence of connections in Section 4 and the weights of those connections in Section 5. Finally, Section 6 discusses how to model the additional heterogeneity in all these features. Throughout the text, links to Section 7 highlight aspects that are particularly challenging or contain pitfalls. Furthermore, links to Section 9 provide URLs to NEST code snippets that help the reader develop an intuition on how the usage of the discussed models would look in code. The chapter ends with some concluding remarks in Section 8.

2 Connectivity

2.1 From empirical data to mathematical models

Anatomical and physiological experiments are yielding ever richer data sets on brain connectivity. Integration of these data into dynamical models can help gain insight into their implications for brain activity and function, besides identifying gaps in the data which can inform future experiments. The data cover diverse scales, ranging from electron microscopy at the sub-micron scale of synapses to light microscopy for neuronal morphology, paired recordings identifying fractions of connected neuron pairs, glutamate uncaging at the scale of tens to hundreds of microns, axonal tracing for long-range connectivity, and diffusion imaging for the whole-brain scale (14).

Despite the richness of the available data, none of these experimental approaches can specify full connectomes at the single-neuron resolution, especially in organisms with complex brain structures such as mammals. Therefore, we need to make predictions in order to complete the detailed connectivity data. One strategy is to find statistical regularities in the existing data and use these to extrapolate to missing data points. Of course, models do not need to be fully data-driven; various abstractions may be used to explore the influence of specific aspects of the connectivity. We illustrate the data-driven approach using the example of the cerebral cortex. In view of the incompleteness of the known cortical connectivity for any individual, we describe the connectivity in a probabilistic manner. A different strategy for generating the connectivity may be to grow connections according to developmental or other plasticity rules (for structural synaptic plasticity, see Section 4).

The cerebral cortex contains different types of excitatory and inhibitory neurons, distinguished by their morphology, electrophysiology, connectivity, and molecular make-up (18, 19). We refer to the set of neurons of the same type in a given cortical area and layer as a *population* (see Fig. 2). Connection probabilities are specific to

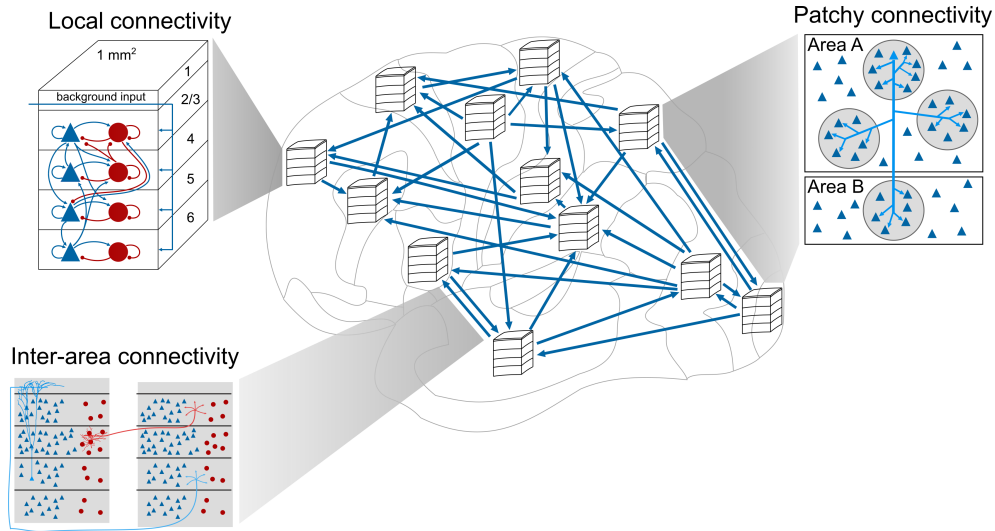


Figure 2: **Substructures of the cortex on different scales.** The neurons in the cortex are organized into areas on the macroscale (middle). In each area, neurons are organized into layers with distinct connectivity (here, shown as a microcircuit under 1 mm² of cortical surface) and into populations of neurons with similar properties within each layer (upper left). Blue triangles and red circles represent excitatory and inhibitory neurons, respectively. Long-range projections connect the areas (lower left). Moreover, on an intermediate scale of millimeters, both intra-area and inter-area excitatory connections cluster into “patches” (right, only outgoing connections for one neuron shown). Adapted with permission from Fig. 1 in Schmidt et al. (15) under license [CC BY 4.0](#) originally from Fig. 1 in Potjans and Diesmann (16) and Fig. 1 in Kunkel et al. (17).

both source and target populations. Both within and between areas, connectivity is also layer-specific. Furthermore, connection probability decays with the distance between neurons, both locally within areas and at longer ranges between areas (20–22). A further organizing principle is that excitatory connectivity tends to form patches (23, 24), meaning that neurons establish additional synapses onto other nearby neurons resulting in spatial clusters (see Fig. 2).

When formalizing these properties into models, a number of subtleties are involved (25). First, the term *connection probability* needs to be defined carefully. This could, for instance, refer to either the total number of synapses divided by the product of the source and target population sizes or the probability for any neuron pair to be connected via at least one synapse. The two definitions diverge in the case of *multapses*, multiple synapses between a given source and target neuron pair, often observed in reconstruction data (26). Further, models can either allow self-connections, also called *autapses*, or prohibit them. Moreover, beyond a certain model size, the

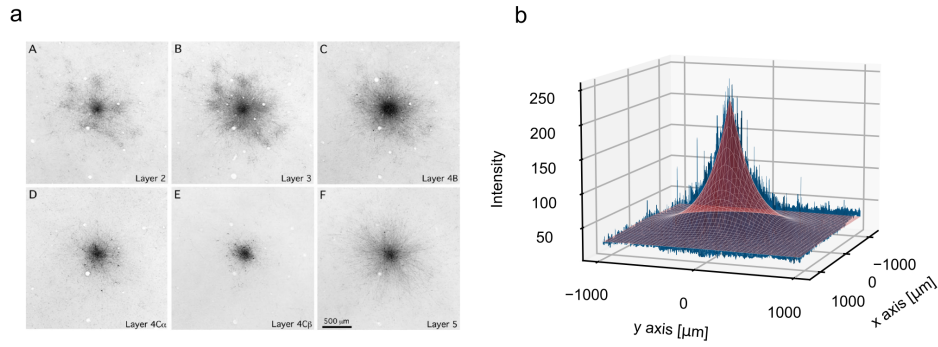


Figure 3: Fitting a model of connectivity to observed data. **a** Layer-resolved axonal tracing data from V1 of New World monkeys . (Adapted from Sincich and Blasdel (27), Fig. 3, Copyright 2001 Society for Neuroscience) Images show staining of cortical layers after an injection of biocytin into layer 3, anterogradely staining axons. **b** Fit of a 2D symmetric exponential function $f(r) = ae^{-\lambda/r} + b$ (red) to the axonal density distribution (blue) of layer 5. The model does not fully capture the connectivity profile for layers 2, 3, and 4B (b) (A, B, C), which display patchy connectivity.

spatial decay of the connection probability becomes important. To capture this, simulated neurons are assigned spatial coordinates, and additional specifications are necessary, including boundary conditions and the choice of connectivity profile. Common choices for the local profile are Gaussian and exponential functions, where the latter generally appears to be a better approximation to experimental data (20, 21).

Fig. 3a illustrates the local decay of connectivity with distance. Choosing a symmetric exponential as a model, Fig. 3b shows that fitting to the experimental data can reveal fundamental constants such as the characteristic length λ .

When including *patchy connectivity*, the spatial position of the patches can be specified via a radial distance from a cell body and an angle (24). Further possible parameters are the number of patches, the size of each patch, and the degree of overlap between patches. Layer-specific axonal tracing data, such as fractions of supragranular labeled neurons from retrograde tracing experiments (28), can inform the laminar inter-area patterns of cortical models. Here one should pay attention to the fact that, on the target side, axonal tracing tells us about axonal or synaptic locations but not about the locations of the target cell bodies. To a reasonable approximation, one can statistically estimate which synapses are established on which target neurons using morphological reconstructions, a method that assigns the number of synapses proportionally to the total length of dendritic elements in the vicinity of the synapses (15, 29).

This is only a tiny selection of data and features that can be included in neuronal

network models. One can go into greater complexity and, for example, consider the higher-level organization of networks, such as hierarchical modularity or small-world properties. For a further discussion on model detail in general, see Section 8.

2.2 From mathematical models to simulation

To simulate how the dynamics of a neuronal network model evolve, the mathematical model description needs to be translated into an executable one. This is preferably done using a dedicated simulator to avoid mistakes in the implementation and to enhance comparability and reproducibility of results (for the precise definition of the different forms of *reproducibility*, see Section 7.2.) Executing a neuronal network simulation typically involves two successive phases: during the *build phase*, the network is set up on the machine by instantiating objects and data structures for neurons and synapses. The subsequent *simulation phase* propagates the network state for a specified biological model time. How fast a simulation runs, i.e., how the biological model time relates to the wall-clock time, crucially depends not only on the machine specifications but also on the representation of the network model on the machine. Parallel computing combines the computational power of many separate compute cores or nodes to enable large-scale simulations; to this end, NEST uses a hybrid approach with the Message Passing Interface (MPI) and Open Multi-Processing (OpenMP). The former enables parallel computing on multiple processors with distributed memory, while the latter enables parallel computing even on single processors with shared memory, referred to as *threading*. The total number of so-called *virtual processes* is determined as the product of the number of MPI processes and the number of OpenMP threads per process. A direct mapping between network structure and hardware is in general difficult to realize. Therefore, NEST uniformly distributes the neurons of each population across the available processors to balance the compute load (Section 7.3). The neurons are connected via synapses, which are assigned specific *weights* and *delays* reflecting conduction times. Synapse models are stored and updated on the same compute nodes that hold their postsynaptic partner neurons. Maintaining the complete network connectivity in computer memory enables the use of plasticity mechanisms that can modify synaptic strengths at runtime (for functional synaptic plasticity, see Section 5). The alternative *procedural connectivity* approach generates the required routing information on the fly and thereby requires fewer memory resources (30, 31).

Establishing synapses in a computational network model requires defining which neurons are connected. For specific data-driven models, the network structure can be

loaded from a file, but simulators also provide built-in routines for generating connectivity. These routines (see Program 1) range from a primitive that just connects individual source and target neurons, to high-level connection rules acting on the neuron population level (25). For example, the deterministic rule *all-to-all* connects each neuron of a source population to each neuron of a target population. Probabilistic rules account for the often statistically described sparse connectivity in biological neuronal networks. *Random, fixed in-degree* connectivity, for instance, specifies only the number of incoming connections per neuron but not which individual ones are selected as sources. If the connectivity is described as *pairwise Bernoulli*, each pair of neurons is connected with a given probability. The fixed in-degree rule needs to be combined with the specification of whether multapses are allowed, whereas the pairwise Bernoulli rule excludes them by definition as each pair of neurons is considered only once. High-level connection rules enable efficient low-level implementations such as parallelization of the network construction.

Pseudo-random number generators (pRNGs) are used for drawing connections according to a probabilistic rule and optionally also for setting neuron and synapse parameters (see Section 6). The resulting network realization will be identical if the same sequence of random numbers is sampled; this is achieved by fixing the pRNG seed. Random distributions sometimes have to be constrained in order to restrict the sign of a weight, e.g., according to Dale's law (32), or to enforce connection delays to be larger than the simulation time step, for which a typical value is 0.1 ms. A longer minimum delay, for instance, 1 ms, can furthermore be used to limit the necessary frequency of communication between virtual processes.

Large-scale neuronal network models require high-performance computing. Employing several compute nodes in parallel not only distributes the workload but also gives access to sufficient memory for storing the network connectivity. Storing a single synaptic weight costs 8 bytes in NEST (33) as it sums up the effects of a set of vesicles that may differ in size, as well as of potentially different amounts of receptors. Moreover, a typical neuron in the mammalian brain has on the order of 10^4 synapses (34). This leads to a substantial amount of resources required for large models. Networks with reduced neuron and synapse numbers can preserve some characteristics (e.g., firing rates) of full-scale networks if the downscaling is compensated for with informed parameter adjustments (35). The pairwise correlation structure of the neuronal activity, however, cannot be preserved simultaneously, rendering neuroscientific simulations at natural density a necessity where correlation structure is relevant. This may be the case, for instance, to ensure the correct network state: correlation changes may even shift a network between linearly stable

and unstable regimes. Data structures that keep the memory usage per MPI process constant regardless of the total number of MPI processes used in the simulation (36) provide a potential solution, paving the way toward brain-size networks with realistic connectivity.

3 Synaptic transmission

3.1 From empirical data to mathematical models

Electrochemical signaling between neurons is mediated by various receptor types, expressed post- and presynaptically. Different receptor types trigger different physiological responses. Ligand- or voltage-gated ion channels influence ionic flows through the membrane, with distinctive kinetics for each receptor type (examples include AMPA and NMDA). Metabotropic receptors trigger intracellular biochemical signaling cascades with downstream actions that are typically not instantaneously noticeable but mediate physiological adaptation processes. *Electrical synapses (gap junctions)* are transmembrane channels that form a direct electrical and biochemical coupling between the cytosol of two adjacent cells. Compared to *chemical synapses*, they provide increased speed as the signal does not need to be converted from electrical to chemical and back across a synaptic cleft. The composition of receptor types on a neuron's synapses, their spatial distribution on the dendritic tree and cell body, and their individual, instantaneous efficacy and response kinetics determine how and at which timescales the neuron filters and integrates its many presynaptic inputs. We will focus here on chemical synapses.

The amplitude of the postsynaptic response is proportional to the synapse's strength or *weight*, which depends on the amount and types of both neurotransmitters and receptors as well as the state of the postsynaptic neuron. Synaptic weights can be estimated from paired recordings, usually *in vitro*, to avoid background activity that confounds the measurements. However, here it should be taken into account that the synaptic weight obtained from paired-cell recordings is determined by a combination of biophysical properties, e.g., postsynaptic receptor density, amount of released neurotransmitters, reuptake kinetics, or existence of more than one connection between a pair of cells (multapses). Hence, the terms *strength* and *weight* refer to an effective, phenomenological quantity. Mapping the corresponding parameters to the *in vivo* condition is nontrivial because experimental conditions like temperature and extracellular fluid may differ, as well as a high-conductance network state affecting the measured quantities such as time constants (37, 38). Change of synaptic

strengths over time is discussed in Section 5.

When a presynaptic neuron has emitted an action potential, the signal that arrives at the postsynaptic neuron can be observed as a postsynaptic potential (PSP): the deflection in the somatic membrane voltage caused by the incoming spike. Alternatively, synaptic currents (PSCs) may be measured at different holding potentials using voltage-clamp recordings. From a PSC or PSP, the weight of the synapse can be derived. Synapses are often modeled as injecting a current into the postsynaptic cell or acting as a conductance, the current of which is proportional to the difference between the membrane potential and a synapse- or receptor-specific reversal potential. In the simplest approximation of the postsynaptic response kinetics, the time course of this current or conductance may be modeled as a Dirac delta function causing a step increase in the membrane potential or current, respectively. With increasing levels of complexity, the time course of the PSC (or postsynaptic conductance, PSG) can be approximated by an instantaneous rise followed by an exponential decay or by a double exponential with separate time constants for the rising and the decaying phase (39).

Beside spatially precise communication via synapses, the spatially more diffuse process of *neuromodulation* can alter the excitability of neurons and affect synaptic plasticity (see Section 5). Neuromodulation is achieved by the release of a neurotransmitter with less detail in the connectivity patterns than in typical synaptic (for instance, glutamatergic) neurotransmission. The neuromodulator, such as dopamine or serotonin, is typically released from a neuron whose cell body lies in a small, circumscribed nucleus in the brain but which projects broadly and affects many downstream targets simultaneously. The precise spatiotemporal profile of neurotransmitter concentration is often approximated in models by assuming the neuromodulator diffuses through extracellular space, referred to as *volume transmission*. Simulating the diffusion process entails solving the Laplace equation—an equation that involves the spatial gradient and divergence operators, requiring a different type of solver than those that solve the neuronal network system dynamics. Instead of a detailed representation of the geometry of extracellular space (for instance, based on the finite-element method), the medium may be assumed to be spatially homogeneous, and diffusion can even be assumed to occur instantaneously, considerably simplifying the model and its computational requirements (40).

Neurons have a spatial extent, and their dendrites often exhibit intricate branching patterns. Consequently, the spatial collocation of synapses on the dendrites has important consequences for the neuron's response to input. Dendritic responses are often nonlinear, as dendrites are studded with a high density of voltage-gated

channels, which, combined with intracellular responses like calcium signaling, can cause a nonlinear interaction between nearby synaptic inputs. In addition, the dendrite itself can exhibit action potentials distinct from a somatic action potential, for instance, involving a local, intracellular calcium transient (see, e.g., 41). The (local) change in membrane potential and conductance, in turn, affects the integration at adjacent synapses in the branch. The triggering of dendritic action potentials by co-activated and co-located synapses and their effects on the somatic dynamics can be accounted for in simple point neuron models by including nonlinearities in synaptic input currents (42, 43). For a more fine-grained analysis, multicompartment models are commonly used. In these models, each neuron consists of dozens or hundreds of compartments, each equipped with a distinct type of dynamics and parameterization and coupled to neighboring compartments according to Ohm's law (39). Multicompartment models permit integrating experimental data at a highly detailed level of description but are computationally and conceptually much more complex and demanding. On the other hand, some biophysical details like synaptic adaptation can be adequately modeled without the need to address the microscopic biophysics of synaptic vesicles but can be treated phenomenologically by adding one or a few extra continuous state variables to the model (e.g., 44, 45).

3.2 From mathematical models to simulation

NEST integrates equations for neurons with linear subthreshold dynamics exactly (46) and uses standard numerical solvers for nonlinear neuron models. For synapses, an efficient approach is to specify the characteristic time evolution of some postsynaptic quantity, such as current or conductance, as a linear system of equations. If responses sum linearly across a neuron's synapses, they can be lumped together into a single or a few state variables and do not have to be stored and updated for each synapse separately. For this reason, the postsynaptic response is typically specified as part of the (postsynaptic) neuron model. As described in Section 3.1, the postsynaptic kernel could be, for example, a Dirac delta function (causing an instantaneous jump in the postsynaptic membrane potential), an instantaneous rise followed by an exponential decay, or a double-exponential function with a finite rise time. Furthermore, because they are linear, solving these equations does not require a numerical solver but only multiplication with a constant at each time step (47). The reduction to a simple multiplication generally makes the solution much more precise: the computed values are closer to the mathematically "true" solution and more efficient to compute. Thus, simulations of networks with many synapses become feasible. Mul-

multiple types of synapses can be easily incorporated into this scheme by grouping them according to their kinetics, for instance, into a separate AMPA and NMDA group (see Program 2).

In simulations of large networks, the layout of data structures in memory and communication can become bottlenecks. Conceptual modeling decisions can interact with data layouts; for example, the synaptic delay can be chosen as a property of the synapses or the pre- or postsynaptic neurons. In the point-neuron framework, the delay is assigned to either a neuron's axonal or dendritic side, implying different biophysical interpretations and simulation outcomes. The biophysical object of a synapse is not necessarily represented in code by a specific software object but distributed into a presynaptic and a postsynaptic component. In the instantiation of a particular model, these components may not even live on the same compute node. As noted in Section 2.2, NEST stores synapses on the process containing the postsynaptic neuron.

In simulations using parallel computing, spike events and potentially other quantities such as synaptic weights have to be communicated between threads, processes, or across a computer network (Section 7.4). Parallel computing presents a set of unique design requirements because the evolution of the dynamical model needs to occur synchronously, lest the model's state becomes internally inconsistent when some parts of it have become desynchronized in time. This requirement can be addressed by instituting a minimum, nonzero transmission delay for each synaptic connection in the model. A delay between the presynaptic spike and the resulting postsynaptic response effectively decouples neurons for this time window so that events can be transmitted across the computer network in a regular cadence at the end of each window (Section 7.5). This decoupling allows simulations to scale to many compute nodes (48).

From a mathematical modeling point of view, gap junctions are much simpler than chemical synapses; their delay is negligible, and they do not filter the input. However, modeling gap junctions numerically can be challenging because they entail an instantaneous coupling between compartments. The *waveform relaxation* technique helps retain simulation efficiency when combining gap junctions with a numerical simulation method that takes advantage of a minimum, nonzero synaptic delay. Each neuron is considered a separate subsystem in this technique, and the gap junction coupling terms (current flowing from one neuron into the other) are solved iteratively. This solution requires exchanging data (in particular, membrane potentials) between the gap-junction coupled neurons only at the end of each minimum delay step, thus limiting the necessary communication frequency. In exchange, it requires only a

modest increase in computation and the size of the communicated packets since solving the forward dynamics of each separately considered neuron needs to be repeated only once per iteration of the waveform relaxation algorithm (49, 50).

4 Structural Plasticity

The models introduced earlier in this chapter had static connectivity (see Section 2). However, macroscopic observations of the brain have revealed that the connections in cortical networks continually change as new synapses form and others dwindle and disappear (for a review, see 51). This rewiring is a lifelong process to encode experiences but happens extensively during development and recovery from lesions in the brain tissue (for a review, see 52). The underlying mechanisms introducing dynamics into the connectivity are summarized as *structural synaptic plasticity*.

4.1 From empirical data to mathematical models

Including structural plasticity mechanisms into a synapse model can increase its biological plausibility, e.g., regarding learning, development, reformation after lesions (53) or topographic map formation (54). Structural plasticity might also be the basis for associative connections (55) and metaplasticity (56). However, plasticity mechanisms capable of generating network connectivity in a principled fashion can also be helpful in other ways. First, they can help reduce dependence on cumbersome and expensive connectivity recordings in animals (see Section 2). Second, they can serve a range of functional purposes. For example, they can enhance learning performance (57) or increase the storage efficiency of long-term memories and, by that, prevent catastrophic forgetting (58). Plasticity mechanisms also frequently serve a homeostatic function. In general, the term “homeostasis” refers to a range of vital physiological processes that assist organisms in maintaining internal states (such as body temperature, blood sugar levels, and heart rate) at optimal levels. Likewise, homeostatic plasticity maintains quantities such as spiking activity or numbers of connections at an energetically or computationally favorable set-point (59). Efficient pruning of the connectivity and preserving sparse connectivity (60) can help save energy and optimize the usage of limited synaptic resources, which is particularly important in neuromorphic computing (57, 61, 62).

To understand the process of developing a comprehensive and accurate mathematical model of structural plasticity, the following paragraphs sketch the steps involved in creating the model suggested by Butz and Van Ooyen (53) as an example. This model is based on the observation that the creation and deletion of synapses

can bring the postsynaptic neuron's firing rate into a certain physiological range. The authors consider synaptic elements, namely axonal boutons on the presynaptic side and dendritic spines on the postsynaptic side. When two such elements are combined, a synapse is created. The dynamics of the number of synaptic elements for each neuron depends on a readiness variable c (associated with the calcium concentration), which indicates the propensity of a neuron to grow synapses. The readiness variable is a low-pass filtered version of the spiking activity and thus approximates the neuron's instantaneous rate up to a scalar multiplier.

The algorithm comprises four steps, which are repeated until the connectivity converges. First, it continuously updates the spiking activity of the neurons since each neuron's mean firing rate influences the creation of synaptic elements. Second, it updates the readiness c for each neuron:

$$\frac{dc}{dt} = -\frac{c(t)}{\tau} + \beta \delta(t - t_j^f), \quad (1)$$

i.e., c decays exponentially with the time constant τ and increases by a fixed amount β whenever the neuron j spikes at t_j^f , where $\delta(\cdot)$ denotes the Dirac delta function and f stands for "firing." Third, a homeostatic rule drives the neuron to reach and maintain a target activity by deleting postsynaptic elements if the instantaneous activity is higher than the target activity or creating synaptic elements if the current activity is lower than the target activity. A growth curve defines the speed of these modifications toward a target calcium concentration c_{target} by means of a growth rate v and can be expressed as a linear function

$$\frac{dz}{dt} = v \left(1 - \frac{c(t)}{c_{\text{target}}} \right). \quad (2)$$

Alternatively, a downward shifted Gaussian or other more complex function may be used, as long as it has a zero-crossing with a negative gradient that allows convergence. If the value of z increases or decreases by 1, the neuron grows or deletes a synaptic element, respectively. The algorithm creates new connections between randomly chosen synaptic elements from the available set in the fourth and last step. This set comprises synaptic elements generated in previous iterations that are not yet connected and the connection partners of deleted synaptic elements.

Beyond this specific example, there exists a range of different structural plasticity models: some have rules for deleting, some for forming synapses, and some for both (for a book, see [63](#)). Often the algorithm prunes synapses that do not have the chance to become active again ([64](#)), are the weakest according to a specific metric

(65, 66), or have too little causal correlation between pre- and postsynaptic spikes (67). Sometimes the mechanism's objective is to maintain a preset number of connections (57), preserve short-range connections (68), or prune until the connectivity has converged to the most efficient constellation (64). In many algorithms, the determining factors for rewiring are the pre- and postsynaptic activity and the vicinity of other synapses. In general, one can divide structural plasticity mechanisms into two categories: Hebbian structural plasticity, which leads to an increase in the number of synapses during phases of high neuronal activity and, conversely, a decrease in phases of low neuronal activity; and homeostatic structural plasticity, which balances these changes by removing and adding synapses (69).

4.2 From mathematical models to simulation

The NEST implementation of the discussed particular structural plasticity mechanism (70) updates the network connectivity at time intervals that are long compared to the computational time step used to update the neurons, based on experimental observations (see Program 3). This slow timescale makes the algorithm more efficient, as the available synaptic elements do not need to be calculated and communicated at every time step. However, when using structural plasticity to generate connections in a network, note that convergence is not guaranteed but determined by the growth rate, network connectivity, and network activity; thus, visual guidance is advised (see 71 and Section 7.6).

5 Functional Plasticity

The strength of a synapse is usually parameterized by a single static value, the synaptic efficacy or synaptic weight (see Section 3). However, existing synapses can grow stronger or weaker as an effect of a variety of biophysical mechanisms on both the pre- and postsynaptic side, phenomena collectively known as *functional synaptic plasticity*. These adjustments of synaptic efficacies are likely to form the basis of learning and memory processes in the brain. Thus, this section addresses the temporal evolution of synaptic efficacies and the underlying mechanisms.

5.1 From empirical data to mathematical models

Over 70 years ago, Hebb famously postulated that neurons that fire together wire together (72). Since then, many phenomenological models of functional plasticity have been derived and developed. Introducing categories brings some order into the

vast landscape of models, even if they do not have clear-cut boundaries and often overlap. Four categorizations are common. First, with respect to the timescale: while short-term plasticity models cover timescales from milliseconds to seconds, long-term plasticity models cover minutes to hours, and homeostatic plasticity models (e.g., synaptic scaling) even up to days (73, 74). While structural plasticity can occur over the course of hours (75), timescales of functional plasticity are typically shorter. One needs structural plasticity to create new synapses (happening on a long timescale) which then grow stronger via functional plasticity (happening on a faster timescale). Vice versa, synapses that have grown weak are more likely to be pruned. To date, models usually include either functional or structural plasticity but not both together, and the effects of these mechanisms on synaptic learning are thus studied independently. A second categorization distinguishes functional plasticity according to the mechanisms involved: Unsupervised learning rules are based on unlabeled data, supervised learning rules involve a target signal, and reinforcement learning rules function via rewards (73, 74). A third categorization considers the number of factors that constitute the update formula of the synaptic efficacy: Standard correlation-based rules usually involve two factors, the pre- and postsynaptic spiking, as opposed to three-factor models that involve an additional modulatory signal, e.g., neuromodulation. Fourth, based on activity dependence, plasticity mechanisms occur in two counteracting forms: in Hebbian-type mechanisms, higher activity levels of the pre- and postsynaptic neurons lead to strengthening of the (positive or negative) weights, whereas in homeostatic mechanisms, to their weakening (69). Without further constraints, Hebbian-type plasticity may lead to a positive feedback loop and, consequently, substantial changes in synaptic weights and network activity. In contrast, homeostatic synaptic plasticity pushes the synaptic efficacy up if activities are low and down if neuronal activities are high, inducing a negative feedback loop and stabilizing the dynamics.

Experimental studies show that the efficacy of a synapse can change for a short time window of hundreds to thousands of milliseconds depending on the history of the presynaptic spikes (76–78). This phenomenon is termed *short-term plasticity (STP)*, or more precisely, *short-term facilitation (STF)* if the efficacy is elevated, and *short-term depression (STD)* if the efficacy is decreased. The biophysical mechanism underlying STP is the dynamics of vesicle pools and spike-triggered exocytosis. On the one hand, after the generation of a spike, calcium accumulates in the presynaptic axon terminal, increasing the probability of neurotransmitter release, which enhances the synaptic efficacy and thus causes STF. On the other hand, repetitive firing leads to the depletion of vesicles and saturation of postsynaptic receptors, which decreases the

efficacy and thus causes STD. The mechanisms for STF and STD are counteracting, and a combination of both can be present in the same synapse. Depending on the synapse or neuron type, one of them may be more pronounced. These phenomena form the basis for many STP models (for a review, see 79). The following paragraph outlines one possible modeling approach by using the example of the model proposed by Tsodyks and Markram (76).

The starting point is the view introduced in Sections 2 and 3: a neuron k receives spikes from neuron j over a synapse with the weight w_{jk} . Now, to make the static synaptic weight a dynamical variable, w_{jk} is multiplied by a time-dependent scaling factor $f_a(t)$, modeled by a set of three coupled differential equations:

$$\frac{df_a}{dt} = -\frac{f_a}{\tau_i} + u^+ f_r^- \delta(t - t_j^f), \quad (3)$$

$$\frac{df_r}{dt} = \frac{f_i}{\tau_r} - u^+ f_r^- \delta(t - t_j^f), \text{ and} \quad (4)$$

$$\frac{df_i}{dt} = \frac{f_a}{\tau_i} - \frac{f_i}{\tau_r}, \quad (5)$$

describing the utilization of synaptic resources by each presynaptic spike arriving at time t_j^f . Here, $f_a(t)$, $f_r(t)$, and $f_i(t) = 1 - f_r(t) - f_a(t)$ denote the fractions of active, recovered, and inactive synaptic resources, respectively, and $\delta(\cdot)$ the Dirac delta function. The superscripts “−” and “+” refer to the values of the associated variables before and after their update. Each presynaptic spike increases the active synaptic resources, i.e., the synaptic weight, and simultaneously reduces the available (recovered) resources by an amount proportional to u^+ . The utilization variable u^+ represents the probability of vesicle release, controlled by the calcium concentration in the axon terminal. In the absence of STF, this utilization is constant. However, in facilitating synapses, it is dynamic and evolves according to

$$\frac{du}{dt} = -\frac{u}{\tau_{fac}} + U(1 - u^-) \delta(t - t_j^f), \quad (6)$$

where the parameter U corresponds to the amplitude of the postsynaptic current (the synaptic weight) in response to a single isolated presynaptic spike. The time constants τ_r , τ_i , and τ_{fac} describe the recovery time from synaptic depression, the decay of the postsynaptic currents, and the decay of the utilization, respectively. Despite its simplicity, this phenomenological model approximates experimental findings well (76).

The modifications of the synaptic efficacy by STP occur only during presynaptic firing and last for a few hundred milliseconds. After presynaptic firing has stopped,

the synaptic resource variables, and hence the synaptic weight, return to their resting states $f_a = 0$, $f_r = 1$, $f_i = 0$, $u = 0$, and w_{jk} . In contrast, *spike-timing-dependent plasticity* (STDP) has a prolonged effect on synaptic efficacy and thus constitutes a form of long-term plasticity. This form of plasticity was discovered in several spike pairing experiments where a pre- and a postsynaptic neuron were repetitively stimulated to emit spikes at a predefined interval (80, 81). Reviews of the experimental findings can be found in Caporale and Dan (82), Markram et al. (83) and Brzosko et al. (84). Although the results of these studies vary across cell types and pairing protocols, they all find that the induced change of the synaptic efficacy depends on the precise time difference $\Delta t = t_{\text{post}} - t_{\text{pre}}$ between a pair of pre- and postsynaptic spikes. The mapping between the sign and value of the weight changes and the time lags between pre- and postsynaptic spikes is described by the *STDP kernel*, which can take either a Hebbian or an anti-Hebbian form and is sometimes modeled as either symmetric or anti-symmetric (all forms illustrated by Fig. 2 in 85). In nature, however, the size and shape of the potentiation and depression windows might differ, leading to an overall asymmetric window (see Fig. 4a). Generally, a postsynaptic spike occurring slightly after the presynaptic spike ($\Delta t > 0$) induces *long-term potentiation* (LTP), whereas a postsynaptic spike occurring slightly before the presynaptic spike ($\Delta t < 0$) induces *long-term depression* (LTD). Thus, STDP can encode a causal relationship between the firing of the pre- and postsynaptic neuron and the synaptic weight change. Since this finding follows Hebb's principle, this type of STDP belongs to the class of Hebbian plasticity rules.

Formalizing this robust finding based on the above experimental data allows for mathematical treatment. Morrison et al. (74) developed a phenomenological model with only a few free parameters, which reproduces the experimental observations without referencing the underlying molecular mechanisms. They restricted the observables that can enter the plasticity rule to locally available ones because biologically plausible phenomenological models should only contain terms that are identifiable with mechanisms that exist in biology. In the case of STDP, the synaptic weight change depends on the pre- and postsynaptic spike times and potentially also on the current synaptic weight. Experiments demonstrating that action potentials back-propagating through the dendritic tree convey information about a postsynaptic spike support the fact that postsynaptic spikes can be available at the synapse (80).

The dependence of LTP and LTD on Δt is usually captured by exponential functions with decay times τ_{\pm} . Morrison et al. (74) give a simple model of the weight

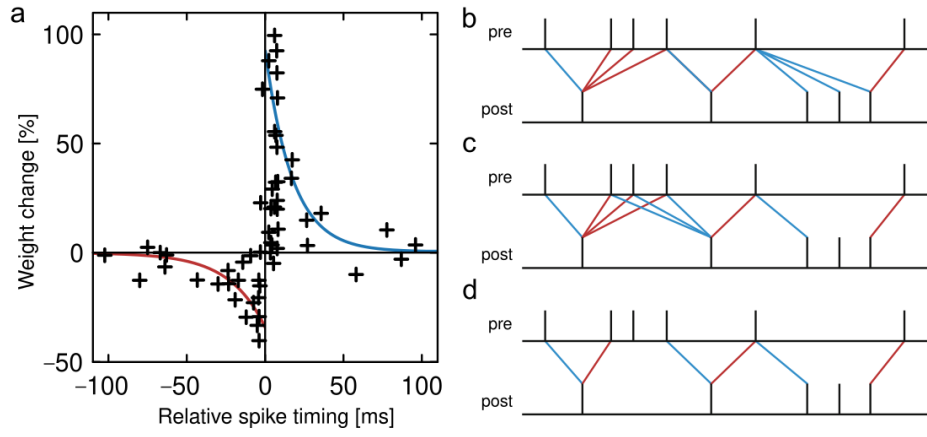


Figure 4: **Spike-timing-dependent plasticity (STDP).** **a** Weight change expressed as a function of relative pre- and postsynaptic spike timing for an example of STDP with an anti-symmetric (and slightly asymmetric), Hebbian learning window. Markers correspond to empirical data from Bi and Poo (81). Solid lines show exponential approximations used in the model (red indicates depression and blue indicates potentiation in all panels). An anti-Hebbian window would look similar but mirrored about the vertical axis. **b-d** Three nearest-neighbor spike pairing rule variants for STDP. **b** *Symmetric*: each presynaptic spike is paired with the last postsynaptic spike, and each postsynaptic spike is paired with the last presynaptic spike. **c** *Presynaptic centered*: each presynaptic spike is paired with the last postsynaptic spike and the next postsynaptic spike. **d** *Reduced symmetric*: as in panel (c), but only for closest pairs. Adapted from Fig. 7 in Morrison et al. (74).

change Δw for pair-based STDP:

$$\begin{aligned} \Delta w_+ &= F_+(w) e^{-\frac{|\Delta t|}{\tau_+}} & \text{if } \Delta t > 0, \\ \Delta w_- &= -F_-(w) e^{-\frac{|\Delta t|}{\tau_-}} & \text{if } \Delta t \leq 0, \end{aligned} \quad (7)$$

where the functions F_{\pm} capture the dependence on the current weight w and have to be specified further by fitting them to experimental data (86, 87) (Fig. 4a). A spike pair can be defined in different ways: for example, each presynaptic spike can be paired with the most recent preceding postsynaptic spike and vice versa, which is one variant of the class of *nearest-neighbor schemes* (see Fig. 4b–d), whereas in the *all-to-all scheme*, each presynaptic spike is paired with all preceding postsynaptic spikes (74, 88).

The STDP model described above can serve as a starting point for designing extended models to describe more nuanced experimental results, for example by including the postsynaptic membrane potential as an additional modulatory factor beyond the pre- and postsynaptic spikes. Along these lines, Clopath and Gerstner

(89) and Clopath et al. (90) account for the effects of voltage-dependent receptors and channels. Their approach is based, among other findings, on experiments showing that the same spike pairing protocol can, depending on the postsynaptic membrane potential V_m , induce no change in synaptic weights at all, LTD, or LTP (91). While a V_m smaller than an experimentally determined threshold potential Θ_- induces neither LTD nor LTP, an intermediate membrane potential $\Theta_- < V_m < \Theta_+$ triggers LTD, and a high $V_m > \Theta_+$ enables LTP. To capture this behavior, the mathematical description of the plasticity rule contains terms for facilitation and depression that are active based on these conditions of the membrane voltage, formally expressed as Heaviside step functions. With this mechanism, Clopath and Gerstner (89) were able to reproduce the complex frequency dependence of the synaptic weight changes in spike pairing experiments (92).

In Urbanczik and Senn (93), the postsynaptic membrane potential is included as a modulating factor. This plasticity rule, in particular, applies to synapses that connect to the dendrite of a postsynaptic neuron. Experiments show that presynaptic spikes that do not cause postsynaptic spikes lead to a depression of synaptic weights whose strength increases with increasing dendritic voltage (94). From this observation, Urbanczik and Senn (93) conclude that the synaptic weights are adjusted such that the dendritic voltage assumes high values if and only if the soma of the postsynaptic neuron emits spikes. Therefore, in this rule, the difference between the dendritic voltage and the somatic activity drives the synaptic weight change.

Instead of the postsynaptic membrane potential, a third factor could also be a neuromodulator concentration, which is motivated by experimental studies (for a review, see 95) and by the fact that they provide a biologically plausible implementation of reward signals (96).

5.2 From mathematical models to simulation

The STP implementation in NEST (see Programs 4 and 5) exploits several practical properties of the corresponding differential equations for their numerical integration. Since the synaptic resources are conserved (i.e., the fractions f_r , f_a , and f_i add up to 1), f_a can be eliminated from the system. Furthermore, thanks to its linear form, the system of coupled differential equations Eqs. (3) to (5) can be integrated exactly between two consecutive presynaptic spikes (46). Concretely, the joint state of u , f_r , and f_a can be iteratively evolved by multiplying the state at the previous presynaptic spike with a *propagator matrix*.

To simulate STDP, Eq. (7) needs to be calculated efficiently (74). Having restricted

the model parameters to those locally available at the synapse facilitates the implementation in software (see Program 6). These constraints also improve the model's performance, since network simulators running on distributed systems take advantage of a limited need for global access to variables to reduce memory consumption and high-latency communication between compute nodes (97, 98). The all-to-all pairing scheme can be efficiently implemented using a specific update scheme of the synaptic traces. These traces represent a fading memory of past spikes at the synapse without explicit knowledge of all past spike times (74, 86). If a pre- or postsynaptic spike occurs, the corresponding trace and synaptic weight are updated, while no actions need to be performed in the periods in between. Defining the exact order of updates in a plasticity model, particularly with regard to pre- and postsynaptic spike timing, indicated by the "+" and "-" in Eqs. (3) to (5) is crucial and facilitated by the high-level language NESTML (see Program 7).

More complex learning scenarios, like reinforcement learning, are made possible by advanced plasticity rules, which, for example, depend on the postsynaptic membrane potential or neuromodulators (99). However, these rules typically make it more difficult to discover an effective implementation. For example, neuromodulators (e.g., dopaminergic signals) affect several nearby synapses through *volume transmission* requiring a notion of physical 3D space (see Section 3.1). Moreover, the presence of time-continuous signals in some of these advanced plasticity rules necessitates the storage of the signal history if one wishes to keep the efficient *event-driven* scheme of updating the synaptic weights only at presynaptic spike times (97). Depending on how the data structures are laid out in a simulator, accessing the continuous third-factor variables can be difficult or computationally costly because they need to be queried for every spike at every synapse. Generally, rules that require only spike times are more efficient in memory and compute time than rules that depend on the entire history of variables, like the membrane potential trace.

Given access to the synaptic weights, another form of functional plasticity, *weight normalization*, can be realized. It entails keeping the total sum, or a norm of all incoming synaptic strengths of a neuron constant by re-normalizing all its synaptic weights (see Program 8). Since, in the brain, these weight changes happen on timescales of several hundreds of milliseconds, the iterative re-normalization takes place on a coarse time grid, increasing the operation's efficiency.

Other advanced plasticity rules include, for example, a third-factor postsynaptic dendritic current (93) or inhibitory plasticity (100). The discovery of new plasticity rules can be, to a certain degree, even automated (101). Furthermore, state models with *synaptic tagging and capture* (STC), described, e.g., in Barrett et al. (102), incorpo-

rate even plasticity effects beyond synapse-specific ones. Ultimately, state-of-the-art computational plasticity models transcend the simple STP and STDP models (see, e.g., [103](#)). Algorithmically, however, these complicated models often use a combination of plasticity mechanisms and thus can be synthesized from such a base stack of simpler models.

6 Heterogeneity

Complexity and heterogeneity are ubiquitous and well-established design principles in neurobiological systems ([104](#)), covering a multitude of components and mechanisms at various spatial and temporal scales. From an information processing perspective, such variability is a fundamental component of the system, as it determines the types of computations a given circuit can perform and constrains the representational expressivity of its dynamics ([105](#)).

6.1 From empirical data to mathematical models

Biological synaptic connectivity is highly diverse in most of its constituent properties, including the type of neurotransmitter used, the composition of presynaptic vesicles and docking proteins (affecting release probability), the postsynaptic receptor composition (affecting efficacy and kinetics of the elicited response), transmitter re-uptake and re-use, and the involvement of gliotransmission (see, e.g., [106](#)), but also properties characterizing signal propagation such as axon diameter and conductance velocity ([107–109](#)). These various types of diversity translate to a high degree of heterogeneity in phenomenological parameters characterizing mathematical models of the synaptic connectivity and dynamics, such as the synaptic weight ([110–114](#)), synaptic time constants ([115, 116](#)), response latencies (synaptic delays; [116, 117](#)), and parameters specifying the plasticity dynamics ([118](#)). In addition, biological neuronal networks exhibit a high degree of heterogeneity in the anatomical connectivity structure, such as the total number of inputs and outputs per neuron (in/out-degrees; [80, 119–123](#)), and the composition of presynaptic source and postsynaptic target neuron populations.

Previous theoretical work on recurrent neuronal networks shows that heterogeneity in single-neuron properties or connectivity broadens the distribution of firing rates ([116, 124](#)) and affects the stability of asynchronous or oscillatory states as well as the level of synchrony ([117, 123, 125–130](#)). A large number of theoretical and experimental studies point at the benefit of heterogeneity for the information processing capabilities of neuronal networks ([105, 129, 131–140](#)). Therefore, modeling studies

aiming at understanding the dynamical and functional principles of biological neuronal networks need to account for the synaptic (and other types of) heterogeneity.

Depending on the type of synaptic heterogeneity, its implementation in mathematical models may follow different strategies. Synaptic heterogeneity is expressed on local scales, such as in the connections between neurons in a given layer of a cortical column, and on large scales, such as in cortical inter-area connections. One form of this heterogeneity results from cell-type, layer, or area specificity. It reflects the anatomical and electrophysiological diversity of neurons in different brain regions and, in addition, emerges from specific interactions with other components of the nervous system or with the environment during brain development and learning. In mathematical models, this specificity is usually accounted for by subdividing the network into several populations representing different cell types or brain regions and applying distinct connectivity, synapse, and plasticity parameters for each pair of populations (Fig. 5a).

Another form of synaptic heterogeneity appears in an unspecific, quasi-random manner. It refers to variations in the synaptic characteristics across an ensemble of neuron pairs of seemingly identical type, for example, connections between a group of neurons with similar morphological and electrophysiological characteristics located in the same layer of a given cortical column (Fig. 5b). Similarly to the cell-type-, layer-, or area-specific diversity described above, the unspecific forms of heterogeneity are partly caused by synaptic plasticity, i.e., by adapting synaptic parameters during learning and development. In this respect, unspecific heterogeneity is not truly unspecific; it is, on the contrary, the result of fine-tuning, optimization, or specialization. Without knowing the details of these processes, the resulting diversity appears random or unspecific. Moreover, the distinction between type-specific and unspecific forms of heterogeneity relies on the assumption that different neuron types are distinguishable (141). Without knowing the characteristics separating two neuronal phenotypes, these cell classes are treated as one type, and the observed diversity in neuron and synapse parameters appears unspecific.

To some extent, synaptic heterogeneity may also result from variations in experimental protocols and in unobserved variables affecting the synapse characteristics. Synaptic weights, for example, are often assessed in voltage-clamp experiments as the amplitudes of somatic postsynaptic currents evoked by presynaptic action potentials. The resulting synaptic weights are then determined not just by the properties of the pre- and postsynaptic cells or by the synapse type and position but also by the holding potential or the electrical characteristics of the electrode-cell contact. Even in the absence of variations in the experimental protocol, the amplitude of the postsynaptic

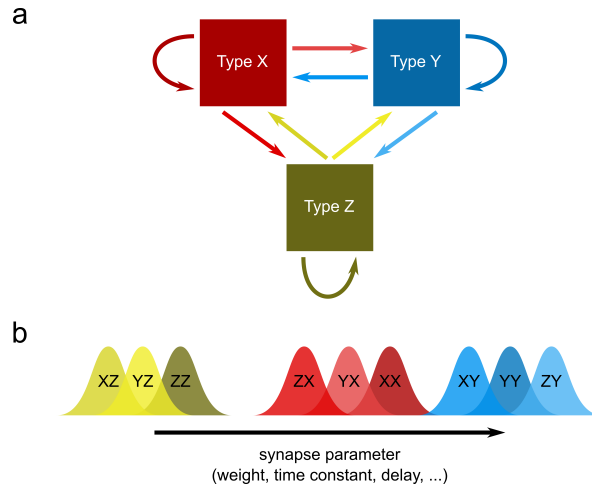


Figure 5: **Specific and unspecific heterogeneity in synaptic connectivity.** **a** Sketch of a neuronal network comprising three populations of neurons of type X , Y , and Z (boxes). The properties of the different projections (arrows), such as the number of synapses, the synaptic weights, synaptic time constants, or synaptic delays, depend on the types of pre- and postsynaptic neurons. We refer to the resulting synapse-type specific diversity as *specific heterogeneity*. **b** For each type of projection $\{PQ\}$ from population Q to population P ($P, Q \in \{X, Y, Z\}$), the synaptic parameters are distributed (illustrated here with bell-shaped curves). We refer to this form of variability as *unspecific heterogeneity*. The parameters characterizing each distribution, such as the mean (horizontal position of each curve) or the variance, are usually synapse-type specific.

response is affected by fluctuations in the postsynaptic membrane potential and by the pre- and postsynaptic spike history. Hidden variables such as the spike history or the synapse position are often not monitored in experimental studies. From the modeler's perspective, it is therefore not straightforward to decide what forms of reported heterogeneity should be accounted for in a given model and what forms are perhaps already represented indirectly by other model features (for example, the voltage dependence of synaptic currents, short-term plasticity, or dendritic filtering in multicompartment models).

In mathematical models, unspecific heterogeneity is typically accounted for in a probabilistic manner. Here, the parameters characterizing synaptic connectivity, such as synaptic weights, time constants, delays, in-degrees, etc., are randomly drawn from certain distributions. In particular, in the brain, many properties follow long-tailed distributions, often approximating the lognormal distribution (142–144). These distributions, or the parameters characterizing them, such as the mean or the standard deviation, are extracted from experimental data. The rationale underlying this

probabilistic modeling approach is twofold. First, it acknowledges that the synaptic parameters are typically not known for every single synapse in a given network. The majority of experimental studies provide data for small subsets of synapses, often pooled across different recording sessions or animals. Second, the probabilistic approach greatly simplifies the models, as the total number of parameters is substantially reduced. In probabilistic modeling approaches, the “model” is not defined by a single instantiation of a network and all its parameters but by the ensemble of many independent realizations generated from a given set of parameter distributions. Observations or findings obtained from a single network realization are meaningless unless they appear generically, i.e., frequently, for many different model realizations.

As described above, synaptic heterogeneity is often the result of an adaptation, development, or fine-tuning process. As demonstrated in a number of studies (see, e.g., 145–147), such processes typically lead to dependencies between parameters. Certain plasticity processes, for example, lead to a competition (anti-correlation) between synapses, such that the strengthening of one synapse results in the weakening of another synapse (148, 149). In a comprehensive probabilistic model, the set of parameters $\{\xi_1, \xi_2, \dots, \xi_n\}$ for a specific network realization is generated according to a joint probability density function (pdf) $p_{1,\dots,n}(x_1, x_2, \dots, x_n)$, describing the probability of observing $\xi_1 = x_1, \xi_2 = x_2, \dots$, and $\xi_n = x_n$. This joint pdf captures all parameter dependencies. Many modeling studies neglect parameter dependencies and assume that the joint pdf $p_{1,\dots,n}(x_1, x_2, \dots, x_n) = p_1(x_1)p_2(x_2) \dots p_n(x_n)$ factorizes. In these studies, each parameter ξ_i is drawn from its respective marginal distribution $p_i(\cdot)$, independently of all other parameters. As before, this simplifying assumption typically reflects a lack of knowledge, as the available experimental data generally do not capture parameter dependencies. Theoretical studies show that this choice can have detrimental consequences for the dynamical and functional properties of the resulting system. Bahuguna et al. (147), for example, demonstrate that when the parameter dependencies are unknown, replacing all parameters by their respective mean (and thereby ignoring diversity) can be a better choice than drawing them from their marginal distributions.

A more direct approach toward modeling synaptic heterogeneity is the explicit account of known plasticity, learning, or developmental processes that dynamically lead to the observed diversity in synaptic parameters, including the dependencies described above (86, 149). Similarly, multivariate parameter distributions may arise from optimization procedures or supervised learning methods fitting the model to some desired target dynamics or behavior (147, 150, 151). While these top-down approaches are promising and commonly used in state-of-the-art computational neu-

rosience, they bear the risk that the underlying data or targets do not sufficiently constrain the model of the actual biological system and hence lead to a multitude of solutions that may not be realized in nature. A combination of bottom-up and top-down constraints appears to be the most reliable method to reduce this form of uncertainty.

6.2 From mathematical models to simulation

Investigating the role of heterogeneity in synaptic connectivity by means of analytical mathematical methods is challenging (116, 117). Therefore, theoretical studies often neglect heterogeneity to simplify the mathematical treatment and provide intuitive insight. A common strategy underlying many mathematical approaches is to reduce the dimensionality of the neuronal network dynamics by assuming that the network can be decomposed into homogeneous subpopulations, each of which comprises neurons with identical neuronal and synaptic parameters. While this approach can account for the specific heterogeneity described above to some extent, it can hardly describe the effects of unspecific heterogeneity. Simulation enables us to test whether the insights obtained under these homogeneity assumptions remain valid if heterogeneity in synaptic parameters is considered.

Even in simulation studies, however, accounting for synaptic heterogeneity is challenging. Acknowledging that every synapse is unique requires representing each synapse with the full individual set of parameters. In a homogeneous network where all synapses have identical properties, the connectivity is fully described by the adjacency matrix (which neuron is connected to which, and how often) and a small set of parameters describing the synapse characteristics, such as the synaptic weight w , the delay d , or the synaptic time constant τ . In a heterogeneous network where each synapse $\{j \rightarrow i\}$ is unique, in contrast, the individual weights w_{ij} , delays d_{ij} , and time constants τ_{ij} need to be stored for each connection. Therefore, representing the heterogeneous connectivity in simulations of neuronal networks at natural density imposes high memory demands for the underlying computing architecture (see Section 7.7).

In models of neuronal networks with heterogeneous synaptic connectivity, the heterogeneity is either implemented by drawing synapse parameters from predefined distributions or by a self-organization process driven by some plasticity or learning dynamics (see Section 6.1). Simulations based on the first, the probabilistic approach, require efficient methods of drawing random numbers from specified distributions during the network generation phase. The NEST simulator, for example,

permits the high-level specification of probabilistic connection rules by the user (see Section 2), including distributions of synaptic weights, synaptic delays, or plasticity parameters. The task of generating a specific connectivity realization by drawing random numbers from these distributions is then delegated to fast low-level (C++) routines. The second approach relies on simulating plastic networks or on numerical optimization methods. Strategies for simulating different forms of synaptic plasticity are described in Sections 4 and 5. Simulating plastic networks with natural connection density is still a major challenge in computational neuroscience. Slow biological processes such as learning and development on timescales of hours, days, and years are presently inaccessible to simulation (or restricted to small and highly simplified models) because of the required wall-clock time. In this respect, dedicated neuro-morphic computing architectures are particularly interesting as simulation platforms for neuroscience, as they offer the potential for faster-than-real-time simulations and hence, for an understanding of plasticity mechanisms on long timescales (152, 153).

7 Notes

In this section, we highlight some of the challenges and pitfalls that may be encountered during modeling and simulation.

7.1 Keeping model refinements biologically plausible

Developing a mathematical model that exhibits a dynamic behavior close to empirical biological data involves iterative optimization procedures, e.g., fitting the experimental data or calibrating the model parameters. While this optimization improves the model in some aspects, it may, at the same time, alter it in ways no longer motivated by biological observations (146). For example, fitting is inherently biased in that, by definition, it improves the validation of certain model features at the cost of those not included in the fitting procedure. Thus, each optimization step should be conducted cautiously and checked for biological plausibility.

7.2 Reproducible simulations

For small-scale simulations, sometimes custom simulation kernels are written. However, besides possibly duplicating published and established routines, these self-made frameworks are likely to contain bugs and lack documentation, for instance, on edge-case behavior. Therefore even for small networks, it helps to use standardized simulators. In particular, these simulators offer the benefit of being well character-

ized under different operating conditions using a diverse array of automated tests, being updated on a regular release cycle, and benefiting from the open-source model of iterative refinement (154). All these factors increase the likelihood of long-term reproducible results.

An individual simulator should exhibit *replicable* behavior: repeated simulations of the same model should yield bitwise identical results, regardless of the number of threads or processing nodes used, due to the use of deterministic pseudo-random number generators. However, simulating the same model on a different platform or using a different numerical solver or time step size for ordinary differential equations (ODEs) may alter the results, especially in network models exhibiting chaotic and unstable dynamics. Nonetheless, results and conclusions should be *reproducible*, obtaining the same overall quantitative and qualitative conclusions (for a commentary on this terminology, see 155). Reproducibility of results requires the original software to be available (including libraries and other dependencies) and, where applicable, the original (raw or pre-processed) dataset(s) and relevant metadata.

Similar to the model descriptions, it increases the reproducibility of methods and results (156) to use and contribute to existing simulation frameworks by reporting bugs, improving implemented methods, and developing and publishing custom modules of the respective framework, e.g., in NEST, in the form of *extension modules* (see Program 9).

7.3 Distribution of compute workload

It is beneficial to distribute the workload evenly across compute nodes, even for networks with complex connectivity and heterogeneous population properties. One way to achieve this is a *round-robin* distribution of neurons across compute nodes, i.e., in the case of M compute nodes, assigning neuron n to node $(n \bmod M)$.

In general, the computing system's size affects the workload distribution. On small machines, the number of synapses per neuron is larger than the number of compute nodes the simulation runs on. Hence, each neuron typically has many targets on every compute node. However, with growing network size and the emergence of new supercomputer architectures over the last decades, the ratio between the numbers of synapses per neuron and compute nodes is in some cases reversed. On these new-generation supercomputers, the distribution of neurons over many nodes decreases the chance that a neuron shares a node with a connected partner, especially considering the sparsity of biological neuronal networks. Mitigating this issue, even more modern compute nodes follow the opposite trend: they possess more memory

and cores per processor and thus more processing power, which reduces the number of nodes required for the simulation and brings the neurons and their targets closer together.

One of the main computational challenges remains the connectivity of neuronal networks. For example, representing each of the estimated 10^{14} synaptic connections in the brain individually by two double-precision numbers requires about 1.6 PB of main memory. Furthermore, neurons form connections with nerve cells not only in their vicinity but also in various remote areas. This feature distinguishes neuronal simulations from simulations of classical physical systems, that use, for example, finite-element methods exploiting the locality of physical interactions. However, memory and communication bandwidth, as well as cache efficiency, are more critical than floating-point performance of spike communication to local and distant targets (157).

Future work should address how computer network connectivity and the simulation distribution across compute nodes can follow (simulated) biological network organization, such as a modular organization on both small and large scales in the brain.

7.4 Scalability in theory and practice

Recording the simulation time under varying network or computing system sizes characterizes an implementation's *scalability*, which is essential to judging its efficiency (36). The scenario of increasing the computing resources while keeping the network size fixed is called *strong scaling*, and the scenario of increasing the network size and computing resources proportionally is called *weak scaling*.

Ideally, when in strong scaling the hardware becomes twice as powerful, the simulation time is divided by two. However, in practice, the gains are usually less due to communication overhead and other bottlenecks in the system. Sometimes, a parallelization can even exhibit a super-linear speed-up (158), but this behavior might only become apparent at very large computing system scales.

Scaling up standard network models to test weak scaling can induce unrealistic activity patterns, for example, regarding the regularity and synchrony of spiking. Since synapses tend to vastly outnumber neurons, the number of synapses is an important determinant of the necessary computing resources. Thus, a reasonable approach is to keep the in-degrees constant when increasing the model size and thereby reduce the overall connection probability (see, e.g., 36). This method tends to lessen activity correlations between neurons and hence diminish synchrony. In

the case of ideal weak scaling, a network of twice the size should run for twice the time, but in practice, the performance is worse due to the same reasons as for strong scaling. Moreover, there is a complex dependence of the scaling behavior on network properties, such as the connectivity's modularity (157).

7.5 Precise spike times in discrete-time simulation

A typical simulation of a continuous-time dynamical process runs in discrete steps of time Δt . However, exchanging events (spikes) on a grid can cause synchronization in the network as a pure simulation artifact. This effect disappears in the limit of $\Delta t \rightarrow 0$, but decreasing the time step increases the time necessary for the simulation to complete, so a tradeoff has to be made. A more computationally efficient solution is to store an extra offset value in spike events, which, in combination with a minimum synaptic delay and an algorithm that finds the precise time of spiking, decouples the simulation time step from the temporal precision with which spikes are exchanged (159).

7.6 Simulating until convergence

Simulating neuroplasticity for too short of a period, especially longer-timescale processes like normalization and neuromodulation, is a common pitfall. Several of these dynamic processes have the potential to cause an abrupt bifurcation in the system late in the simulation. Additionally, some plasticity rules produce a long-tailed distribution of synaptic strengths (160), whereby the distribution reaches equilibrium again only after an extended simulation period. A limited measurement duration can also be a problem in empirical neuroscience, but simulations can, in principle, run as long as desired. The only limitations are the computing resources available (as these need to be shared with other users on high-performance computing systems) and the amount of time the simulation takes to complete, which depends on the simulator's efficiency. Simulating until the measure of learning performance has adequately converged can circumvent this pitfall to some degree.

7.7 Limited synaptic weight resolution

Limiting the numerical resolution of synaptic parameters, such as the synaptic weight, appears to be an obvious strategy to reduce memory and compute load. To some extent, nature itself copes well with quantized synaptic weights: Transmission in chemical synapses is quantized due to the release of neurotransmitters in discrete packages from vesicles in the presynaptic axon terminals. The analysis of

spontaneous (miniature) postsynaptic currents, i.e., postsynaptic responses to the release of neurotransmitters from single presynaptic vesicles, reveals that the resolution of synaptic weights is indeed finite for chemical synapses. As shown by Malkin et al. (161), the amplitudes of spontaneous excitatory postsynaptic currents recorded from different types of excitatory and inhibitory cortical neurons follow a unimodal distribution with a peak at about 20 pA and a lower bound at about 10 pA. Such a cut-off is present despite several factors that may wash out the discreteness of the synaptic transmission, such as variability in vesicle sizes, variability in the position of vesicle fusion zones, quasi-randomness in neurotransmitter diffusion across the synaptic cleft, and variability in postsynaptic receptor densities. However, the discreteness of synaptic strengths is obscured for evoked synaptic responses involving neurotransmitter release from many presynaptic vesicles and for superpositions of inputs from many synapses, and thus unlikely to play a particular role in the dynamics of the neuronal network as a whole.

Inspired by these observations, Dasbach et al. (162) systematically investigated the effects of a limited synaptic weight resolution on the dynamics of recurrent spiking neuronal networks resembling local cortical circuits. They show that a naive quantization of synaptic weights generally leads to a distortion of the firing statistics. However, in the example of one network type, they could demonstrate that the firing statistics remain unaffected under a weight discretization that preserves the mean and variance of the total synaptic input currents. In networks with sufficiently heterogeneous in-degrees, the firing statistics stay constant, even when replacing all synaptic weights with the mean of the weight distribution, i.e., entirely neglecting the unspecific form of heterogeneity in synaptic weights. Applying this finding in simulations reduces the memory demands substantially. The effect of discretized synaptic weights in networks undergoing different forms of synaptic plasticity has rarely been investigated (163) and remains a subject for future study.

7.8 Precise specification of the model

An incomplete and ambiguous description of a model without following conventions can impede understanding by the reader and thus the reproducibility of the respective study. The first step to avoiding this pitfall is to gain a clear picture of the requirements a model description must fulfill. For this, Nordlie et al. (164) propose that a neuronal network model description must contain a complete and detailed account of its architecture and the dynamics of its parts. In the formalization of model descriptions, computational neuroscience is less mature than other fields of

science. Nonetheless, best practice guidelines are emerging that suggest the use of standardized tables of the model characteristics that cover the network architecture and connectivity, all neuron and synapse models used, the applied input stimuli, and the recorded data, described by a combination of text, equations, figures, subtables, and pseudocode (164). Formalized sketches of the network and unified connectivity concepts (25) help computational neuroscientists to unequivocally convey their models and, consequently, readers to understand them.

It is advisable to use one of the numerous formal languages that facilitate such specifications and make them consistent, e.g., NeuroML (13), NESTML (6), or PyNN (12). Most of such languages adhere to the class of either declarative or procedural languages. While a declarative language specifies the model's features, a procedural language specifies the series of commands or instructions needed for constructing the model. Best practices for either of these approaches include formally defined syntax and semantics or an API specification, both uniquely identified by version numbers.

It is good practice to keep the specification and implementation of a model separate. For example, implementation details such as the time resolution and the spike threshold detection method are essential for the reproducibility of the results but are not part of the model itself. Like the model description, the implementation specification should be complete and sufficient, as it can be challenging to reverse-engineer implementations and test the robustness of the results to implementation alterations (164). However, the model should be robust to different choices in such implementation details.

Generating an executable representation of the model can be fully automated. For example, tools like NESTML specialize in processing the model descriptions of different complexity levels and in verbose formats like XML, creating executable implementations and visualizations, and facilitating debugging (165).

To summarize, it is advisable to follow standardized model descriptions, implementations, and generation procedures wherever possible for a project and ideally share the models with the community in dedicated databases for computational neuroscience models like ModelDB (166) or Open Source Brain (167).

8 Conclusions

Large-scale neuronal network simulations are a key tool for understanding brain processes. They make the complex nonlinear dynamics of neuronal activity, which are out of reach with analytic methods, accessible to inquiry. Moreover, continuous in-

teraction between computational modeling and advances in empirical understanding have iteratively refined simulation approaches throughout the history of computational neuroscience. We hence outline a few promising directions for empirical and modeling approaches to work together.

New empirical data can bolster simulation studies in many ways. For example, long-term tracking of synapses *in vivo* will elucidate the relationship between synaptic plasticity and function (168). Besides, recordings of complete connectomes at single-neuron resolution are becoming feasible for ever-larger brains and may soon be available on the scale of a mouse brain (169). Data-driven models based on such detailed and specific connectivity are complemented by models whose connectivity is generated with statistical approaches. The latter models also profit from more available data constraining their parameters; higher-level network organization can, for instance, be informed by data on hierarchical modularity and small-world properties. Sometimes, the conditions of data retrieval are inconsistent between experiments, or experiments only cover a small subset of the model system, and modeling could benefit from additional studies to fill the gaps.

How the available data are integrated into models depends on the particular research question and the level of abstraction appropriate for it. The modeler needs to decide (or find out) which features and phenomena of the natural system need to be represented in detail and which ones can be approximated. For instance, a biophysically detailed model with discrete vesicle release dynamics would be suitable when investigating how a compound influencing vesicle fusion to the membrane affects synaptic transmission. In contrast, investigating the compound's effects on large-scale network dynamics could necessitate approximation of the vesicle release by a simplified set of continuous quantities and differential equations. In essence, a good computational model should represent the relevant attributes of the studied biological structure, have explanatory power and simultaneously not necessitate extensive simulation time.

There are several ways in which the infrastructure of computer simulations itself can innovate, indicated by current trends and feature requests from the community. Simulation efficiency is still a bottleneck, especially in simulations involving plasticity, as they need to run for a comparatively long time. Furthermore, large-scale networks require powerful, high-performance compute clusters, which provide large amounts of RAM, and enable running the simulation in parallel and distributed across many CPUs or GPUs which are interconnected through a low-latency network. More advanced synaptic plasticity rules, for example involving tripartite synapses influenced by astrocytes or neuromodulators like dopamine, still lack software support for effi-

cient simulations on a larger scale. Finally, simulations at cellular resolution could be extended toward multiphysics modeling by incorporating other physical phenomena. Such models could account for the volume diffusion of neuromodulators or the electric field in the neuropil to simulate ephaptic coupling for instance.

In general, what should one strive for in a model? The statistician George Box famously said, “All models are wrong, but some are useful.” Although it helps to remind ourselves of the difficulties of modeling, to say that all models are wrong is not doing them justice, as put by Sir David R. Cox in a comment on Chatfield (170): “The very word *model* implies simplification and idealization. The idea that complex physical, biological or sociological systems can be exactly described by a few formulae is patently absurd. The construction of idealized representations that capture important stable aspects of such systems is, however, a vital part of general scientific analysis and statistical models [...]”. A model with unlimited parameters can fit the data perfectly but at the cost of generalizability and explanatory power. Ockham’s razor, or the law of parsimony, provides a helpful heuristic in this context, guiding us to prefer the theory with fewer parameters between two competing theories. This objective is also formalized in Bayesian information criteria, which penalize models with larger numbers of parameters. In other words, the aim to reduce complexity should guide modeling choices to address a given research question.

However, condensing the biophysical details and terminology to arrive at a parsimonious, phenomenological formulation could impede testing such a minimal model experimentally. Moreover, the rigorous application of Ockham’s razor leads to models optimized for single phenomena (e.g., connectivity, synaptic transmission, structural or functional plasticity) that are thus hard to combine. To solve this problem, after reducing the phenomena to their essential variables, we should express each model in a way that allows for their combination. Biophysical details could provide contact points between model concepts from different subdomains. Progress in integrating heterogeneous phenomena in large-scale models requires that models act as platforms that can be modified and extended over time. For example, after a minimal model has achieved a satisfactory performance, modelers could test whether the same results still hold for models of greater biophysical detail.

As a foreseeable trend, more and more computational neuroscientists will adopt procedures ensuring the reproducibility of their methods and results over the following years. This approach includes, where possible, data, model, and code sharing in open-access online repositories, adherence to open standards for model formats and software tools, active management of metadata, containerized distribution of depen-

dencies, unit testing, and continuous integration instead of creating new in-house toolchains from the ground up. Last but not least, active contribution to an existing model database or open-source software, be it as small as a feature request, is a low-threshold action conducive to reproducible research everyone can take. Ultimately, the community's research interests in the form of these requests and contributions shape the landscape of available models in an open-access simulator.

Acknowledgments

The authors thank Dennis Terhorst and Jessica Mitchell for technical help with the NEST documentation links, Angela Fischer for advice on the layout of the figures, and Rainer Waser for helpful feedback on the manuscript. Furthermore, they gratefully acknowledge funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under Specific Grant Agreement No. 945539 (Human Brain Project SGA3), from the Joint Lab "Supercomputing and Modeling for the Human Brain," the Helmholtz Association Initiative and Networking Fund under project number SO-092 (Advanced Computing Architectures), the Federal Ministry of Education and Research BMBF under grant ID 03ZU1106CB within NeuroSys as part of the initiative "Clusters4Future," the Helmholtz Metadata Collaboration (HMC) under funding ID ZT-I-PF-3-026, the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant ID 368482240/GRK2416, HiRSE_PS, the Helmholtz Platform for Research Software Engineering – Preparatory Study, an innovation pool project of the Helmholtz Association, and the Jülich-Aachen Research Alliance Center for Simulation and Data Science (JARA-CSD) School for Simulation and Data Science (SSD), as well as computing time granted by the JARA Vergabegremium and provided on the JARA Partition part of the supercomputer JURECA DC at Forschungszentrum Jülich (computation grant JINB33).

References

- [1] Lidia Alonso-Nanclares, Juncal Gonzalez-Soriano, JR Rodriguez, and Javier DeFelipe. Gender differences in human cortical synaptic density. *Proceedings of the National Academy of Sciences*, 105(38):14615–14619, 2008. URL <https://doi.org/10.1073/pnas.0803652105>.
- [2] David J Linden. Our human brain was not designed all at once by a genius inventor on a blank sheet of paper. In *Think Tank: Forty Neuroscientists Explore the Biological Roots of Human Experience*, pages 1–8. Yale University Press, 2018. URL <https://doi.org/10.12987/9780300235470-002>.
- [3] Frederico AC Azevedo, Ludmila RB Carvalho, Lea T Grinberg, José Marcelo Farfel, Renata EL Ferretti, Renata EP Leite, Wilson Jacob Filho, Roberto Lent, and Suzana Herculano-Houzel. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, 513(5):532–541, 2009. URL <https://doi.org/10.1002/cne.21974>.
- [4] Guido Trenscho, Robin Gutzen, Inga Blundell, Michael Denker, and Abigail Morrison. Rigorous neural network simulations: A model substantiation methodology for increasing the correctness of simulation results in the absence of experimental validation data. *Frontiers in Neuroinformatics*, 12, 2018. URL <https://doi.org/10.3389/fninf.2018.00081>.
- [5] Marc-Oliver Gewaltig and Markus Diesmann. NEST (NEural Simulation Tool). *Scholarpedia*, 2(4):1430, 2007. URL <https://doi.org/10.4249/scholarpedia.1430>.
- [6] Dimitri Plotnikov, Bernhard Rumpe, Inga Blundell, Tammo Ippen, Jochen Martin Eppler, and Abigail Morrison. NESTML: a modeling language for spiking neurons. *arXiv*, 1606.02882, 2016. URL <https://doi.org/10.48550/arXiv.1606.02882>.
- [7] Michael L Hines and Nicholas T Carnevale. NEURON: a tool for neuroscientists. *The Neuroscientist*, 7(2):123–135, 2001. URL <https://doi.org/10.1177/107385840100700207>.
- [8] Marcel Stimberg, Romain Brette, and Dan FM Goodman. Brian 2, an intuitive and efficient neural simulator. *eLife*, 8:e47314, 2019. URL <https://doi.org/10.7554/eLife.47314>.
- [9] Trevor Bekolay, James Bergstra, Eric Hunsberger, Travis DeWolf, Terrence C Stewart, Daniel Rasmussen, Xuan Choo, Aaron Voelker, and Chris Eliasmith.

- Nengo: a python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*, 7:48, 2014. URL <https://doi.org/10.3389/fninf.2013.00048>.
- [10] N Abi Akar, B Cumming, V Karakasis, A Küsters, W Klijn, A Peyser, and S Yates. Arbor — A Morphologically-Detailed Neural Network Simulation Library for Contemporary High-Performance Computing Architectures. In *27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 274–282, 2019. URL <https://doi.org/10.1109/EMPD.P.2019.8671560>.
- [11] Julien Vitay, Helge Ülo Dinkelbach, and Fred H Hamker. ANNarchy: a code generation approach to neural simulations on parallel hardware. *Frontiers in Neuroinformatics*, 9:19, 2015. URL <https://doi.org/10.3389/fninf.2015.00019>.
- [12] Andrew P Davison, Daniel Brüderle, Jochen M Eppler, Jens Kremkow, Eilif Muller, Dejan Pecevski, Laurent Perrinet, and Pierre Yger. Pynn: a common interface for neuronal network simulators. *Frontiers in Neuroinformatics*, 2:11, 2009. URL <https://doi.org/10.3389/neuro.11.011.2008>.
- [13] Padraig Gleeson, Sharon Crook, Robert C. Cannon, Michael L. Hines, Guy O. Billings, Matteo Farinella, Thomas M. Morse, Andrew P. Davison, Subhasis Ray, Upinder S. Bhalla, Simon R. Barnes, Yoana D. Dimitrova, and R. Angus Silver. NeuroML: A language for describing data driven models of neurons and networks with a high degree of biological detail. *PLOS Computational Biology*, 6, 2010. URL <https://doi.org/10.1371/journal.pcbi.1000815>.
- [14] Sacha Jennifer van Albada, Aitor Morales-Gregorio, Timo Dickscheid, Alexandros Goulas, Rembrandt Bakker, Sebastian Bludau, Günther Palm, Claus-Christian Hilgetag, and Markus Diesmann. Bringing anatomical information into neuronal network models. In M Giugliano, M Negrello, and D Linaro, editors, *Computational Modelling of the Brain*, volume 1359 of *Advances in Experimental Medicine and Biology*, pages 201–234. Springer, 2021. URL https://doi.org/10.1007/978-3-030-89439-9_9.
- [15] Maximilian Schmidt, Rembrandt Bakker, Claus C Hilgetag, Markus Diesmann, and Sacha J van Albada. Multi-scale account of the network structure of macaque visual cortex. *Brain Structure and Function*, 223(3):1409–1435, 2018. URL <https://doi.org/10.1007/s00429-017-1554-4>.
- [16] Tobias C Potjans and Markus Diesmann. The cell-type specific cortical micro-

- p>circuit: relating structure and activity in a full-scale spiking network model.
- Cerebral Cortex*
- , 24(3):785–806, 2014.
- [17] Susanne Kunkel, Tobias C Potjans, Abigail Morrison, and Markus Diesmann. Simulating macroscale brain circuits with microscale resolution. In *Frontiers in Neuroinformatics*, 2009. URL <https://doi.org/10.3389/conf.neuro>.
 - [18] Nathan W Gouwens, Staci A Sorensen, Jim Berg, Changkyu Lee, Tim Jarsky, Jonathan Ting, Susan M Sunkin, David Feng, Costas A Anastassiou, Eliza Barkan, et al. Classification of electrophysiological and morphological neuron types in the mouse visual cortex. *Nature Neuroscience*, 22(7):1182–1195, 2019. URL <https://doi.org/10.1038/s41593-019-0417-0>.
 - [19] Rebecca D Hodge, Trygve E Bakken, Jeremy A Miller, Kimberly A Smith, Eliza R Barkan, Lucas T Graybuck, Jennie L Close, Brian Long, Nelson Johansen, Osnat Penn, et al. Conserved cell types with divergent features in human versus mouse cortex. *Nature*, 573(7772):61–68, 2019. URL <https://doi.org/10.1038/s41586-019-1506-7>.
 - [20] Adam M Packer and Rafael Yuste. Dense, unspecific connectivity of neocortical parvalbumin-positive interneurons: a canonical microcircuit for inhibition? *Journal of Neuroscience*, 31(37):13260–13271, 2011. URL <https://doi.org/10.1523/JNEUROSCI.3131-11.2011>.
 - [21] Rodrigo Perin, Thomas K Berger, and Henry Markram. A synaptic organizing principle for cortical neuronal groups. *Proceedings of the National Academy of Sciences*, 108(13):5419–5424, 2011. URL <https://doi.org/10.1073/pnas.1016051108>.
 - [22] Mária Ercsey-Ravasz, Nikola T Markov, Camille Lamy, David C van Essen, Kenneth Knoblauch, Zoltán Toroczkai, and Henry Kennedy. A predictive network model of cerebral cortical connectivity based on a distance rule. *Neuron*, 80(1):184–197, 2013. URL <https://doi.org/10.1016/j.neuron.2013.07.036>.
 - [23] Daniel J Felleman and David C van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1):1–47, 1991. URL <https://doi.org/10.1093/cercor/1.1.1-a>.
 - [24] Nicole Voges, Almut Schüz, Ad Aertsen, and Stefan Rotter. A modeler’s view on the spatial structure of intrinsic horizontal connectivity in the neocortex. *Progress in Neurobiology*, 92(3):277–292, 2010. URL <https://doi.org/10.1016/j.pneurobio.2010.05.001>.

- [25] Johanna Senk, Birgit Kriener, Mikael Djurfeldt, Nicole Voges, Han-Jia Jiang, Lisa Schüttler, Gabriele Gramelsberger, Markus Diesmann, Hans E Plesser, and Sacha J van Albada. Connectivity concepts in neuronal network modeling. *PLOS Computational Biology*, 18(9):e1010086, 2022. URL <https://doi.org/10.1371/journal.pcbi.1010086>.
- [26] Narayanan Kasthuri, Kenneth Jeffrey Hayworth, Daniel Raimund Berger, Richard Lee Schalek, José Angel Conchello, Seymour Knowles-Barley, Dongil Lee, Amelio Vázquez-Reina, Verena Kaynig, Thouis Raymond Jones, et al. Saturated reconstruction of a volume of neocortex. *Cell*, 162(3):648–661, 2015. URL <https://doi.org/10.1016/j.cell.2015.06.054>.
- [27] Lawrence C. Sincich and Gary G. Blasdel. Oriented axon projections in primary visual cortex of the monkey. *Journal of Neuroscience*, 21(12):4416–4426, 2001. URL <https://doi.org/10.1523/JNEUROSCI.21-12-04416.2001>.
- [28] Nikola T Markov, Julien Vezoli, Pascal Chameau, Arnaud Falchier, René Quilodran, Cyril Huissoud, Camille Lamy, Pierre Misery, Pascale Giroud, Shimon Ullman, et al. Anatomy of hierarchy: feedforward and feedback pathways in macaque visual cortex. *Journal of Comparative Neurology*, 522(1):225–259, 2013. URL <https://doi.org/10.1002/cne.23458>.
- [29] Christopher L Rees, Keivan Moradi, and Giorgio A Ascoli. Weighing the evidence in Peters’ rule: does neuronal morphology predict connectivity? *Trends in Neurosciences*, 40(2):63–71, 2017. URL <https://doi.org/10.1016/j.tins.2016.11.007>.
- [30] Ulrich Roth, Frank Eckardt, Axel Jahnke, and Heinrich Klar. Efficient on-line computation of connectivity: Architecture of the connection unit of NESPINN. In *Proceedings of the MicroNeuro*, pages 31–39, Dresden, 1997.
- [31] James C Knight and Thomas Nowotny. Larger GPU-accelerated brain simulations with procedural connectivity. *Nature Computational Science*, 1(2):136–142, 2021. URL <https://doi.org/10.1038/s43588-020-00022-7>.
- [32] Piergiorgio Strata and Robin Harvey. Dale’s principle. *Brain Research Bulletin*, 50(5-6):349–350, 1999. URL [https://doi.org/10.1016/S0361-9230\(99\)00100-8](https://doi.org/10.1016/S0361-9230(99)00100-8).
- [33] Susanne Kunkel, Maximilian Schmidt, Jochen M Eppler, Hans E Plesser, Gen Masumoto, Jun Igarashi, Shin Ishii, Tomoki Fukai, Abigail Morrison, Markus Diesmann, et al. Spiking network simulation code for petascale computers.

- Frontiers in Neuroinformatics*, 8:78, 2014. URL <https://doi.org/10.3389/fninf.2014.00078>.
- [34] Eric R Kandel, James H Schwartz, Thomas M. Jessell, Steven A Siegelbaum, and A J Hudspeth. *Principles of neural science*, page 1472. McGraw-Hill, 5th edition, 2012.
- [35] Sacha Jennifer van Albada, Moritz Helias, and Markus Diesmann. Scalability of asynchronous networks is limited by one-to-one mapping between effective connectivity and correlations. *PLOS Computational Biology*, 11(9):e1004490, 2015. URL <https://doi.org/10.1371/journal.pcbi.1004490>.
- [36] Jakob Jordan, Tammo Ippen, Moritz Helias, Itaru Kitayama, Mitsuhsa Sato, Jun Igarashi, Markus Diesmann, and Susanne Kunkel. Extremely scalable spiking neuronal network simulation code: From laptops to exascale computers. *Frontiers in Neuroinformatics*, 12:2, 2018. URL <https://doi.org/10.3389/fninf.2018.00002>.
- [37] Alain Destexhe, Michael Rudolph, and Denis Paré. The high-conductance state of neocortical neurons in vivo. *Nature Reviews Neuroscience*, 4(9):739–751, 2003. URL <https://doi.org/10.1038/nrn1198>.
- [38] Andrei Maksimov, Markus Diesmann, and Sacha J van Albada. Criteria on balance, stability, and excitability in cortical networks for constraining computational models. *Frontiers in Computational Neuroscience*, 12:44, 2018. URL <https://doi.org/10.3389/fncom.2018.00044>.
- [39] Wulfram Gerstner, Werner M. Kistler, Richard Naud, and Liam Paninski. *Neuronal Dynamics: from single neurons to networks and models of cognition*. Cambridge University Press, 2014. URL <https://doi.org/10.1017/CBO9781107447615>.
- [40] Wiebke Potjans, Abigail Morrison, and Markus Diesmann. Enabling functional neural circuit simulations with distributed computing of neuromodulated plasticity. *Frontiers in Computational Neuroscience*, 4:141, 2010. URL <https://doi.org/10.3389/fncom.2010.00141>.
- [41] Matthew E. Larkum, Jiameng Wu, Sarah A. Duverdin, and Albert Gidon. The Guide to Dendritic Spikes of the Mammalian Cortex In Vitro and In Vivo. *Neuroscience*, 489:15–33, 2022. URL <https://doi.org/10.1016/j.neuroscience.2022.02.009>.

- [42] Sven Jahnke, Marc Timme, and Raoul-Martin Memmesheimer. Guiding synchrony through random networks. *Physical Review X*, 2(4):041016, 2012. URL <https://doi.org/10.1103/physrevx.2.041016>.
- [43] Younes Bouhadjar, Dirk J Wouters, Markus Diesmann, and Tom Tetzlaff. Sequence learning, prediction, and replay in networks of spiking neurons. *PLOS Computational Biology*, 18(6):e1010233, 2022. URL <https://doi.org/10.1371/journal.pcbi.1010233>.
- [44] Misha Tsodyks, Klaus Pawelzik, and Henry Markram. Neural networks with dynamic synapses. *Neural Computation*, 10(4):821–835, 1998. URL <https://doi.org/10.1162/089976698300017502>.
- [45] Romain Brette and Wulfram Gerstner. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of Neurophysiology*, 94(5):3637–3642, 2005. URL <https://doi.org/10.1152/jn.00686.2005>.
- [46] Stefan Rotter and Markus Diesmann. Exact digital simulation of time-invariant linear systems with applications to neuronal modeling. *Biological Cybernetics*, 81(5-6):381–402, 1999. URL <https://doi.org/10.1007/s004220050570>.
- [47] Abigail Morrison, Sirko Straube, Hans Ekkehard Plesser, and Markus Diesmann. Exact subthreshold integration with continuous spike times in discrete-time neural network simulations. *Neural Computation*, 19(1):47–79, 2007. URL <https://doi.org/10.1162/neco.2007.19.1.47>.
- [48] Abigail Morrison and Markus Diesmann. Maintaining causality in discrete time neuronal network simulations. In *Lectures in Supercomputational Neurosciences: Dynamics in Complex Brain Networks*, pages 267–278. Springer, 2007. URL https://doi.org/10.1007/978-3-540-73159-7_10.
- [49] Jan Hahne, Moritz Helias, Susanne Kunkel, Jun Igarashi, Matthias Boltz, Andreas Frommer, and Markus Diesmann. A unified framework for spiking and gap-junction interactions in distributed neuronal network simulations. *Frontiers in Neuroinformatics*, 9:22, 2015. URL <https://doi.org/10.3389/fninf.2015.00022>.
- [50] Jakob Jordan, Moritz Helias, Markus Diesmann, and Susanne Kunkel. Efficient communication in distributed simulations of spiking neuronal networks with gap junctions. *Frontiers in Neuroinformatics*, 14:12, 2020. URL <https://doi.org/10.3389/fninf.2020.00012>.

- [51] Dan D Stettler, Homare Yamahachi, Wu Li, Winfried Denk, and Charles D Gilbert. Axons and synaptic boutons are highly dynamic in adult visual cortex. *Neuron*, 49(6):877–887, 2006. URL <https://doi.org/10.1016/j.neuron.2006.02.018>.
- [52] Markus Butz, Florentin Wörgötter, and Arjen van Ooyen. Activity-dependent structural plasticity. *Brain Research Reviews*, 60(2):287–305, 2009. URL <https://doi.org/10.1016/j.brainresrev.2008.12.023>.
- [53] Markus Butz and Arjen van Ooyen. A simple rule for dendritic spine and axonal bouton formation can account for cortical reorganization after focal retinal lesions. *PLOS Computational Biology*, 9(10):e1003259, 2013. URL <https://doi.org/10.1371/journal.pcbi.1003259>.
- [54] Simeon A Bamford, Alan F Murray, and David J Willshaw. Synaptic rewiring for topographic mapping and receptive field development. *Neural Networks*, 23(4):517–527, 2010. URL <https://doi.org/10.1016/j.neunet.2010.01.005>.
- [55] Júlia V Gallinaro and Stefan Rotter. Associative properties of structural plasticity based on firing rate homeostasis in recurrent neuronal networks. *Scientific Reports*, 8(1):1–13, 2018. URL <https://doi.org/10.1038/s41598-018-22077-3>.
- [56] Georgios Kalantzis and Harel Z Shouval. Structural plasticity can produce metaplasticity. *PLOS One*, 4(11):e8062, 2009. URL <https://doi.org/10.1371/journal.pone.0008062>.
- [57] Guillaume Bellec, David Kappel, Wolfgang Maass, and Robert Legenstein. Deep rewiring: Training very sparse deep networks. *arXiv*, 1711.05136, 2017. URL <https://doi.org/10.48550/arXiv.1711.05136>.
- [58] Andreas Knoblauch. Impact of structural plasticity on memory formation and decline. In *The rewiring brain*, pages 361–386. Elsevier, 2017. URL <https://doi.org/10.1016/B978-0-12-803784-3.00017-2>.
- [59] G. Turrigiano. Homeostatic synaptic plasticity: Local and global mechanisms for stabilizing neuronal function. *Cold Spring Harbor Perspectives in Biology*, 4(1):a005736–a005736, 2012. URL <https://doi.org/10.1101/cshperspect.a005736>.
- [60] David Kappel, Stefan Habenschuss, Robert Legenstein, and Wolfgang Maass. Network plasticity as Bayesian inference. *PLOS Computational Biology*, 11(11):e1004485, 2015. URL <https://doi.org/10.1371/journal.pcbi.1004485>.

- [61] Sebastian Billaudelle, Benjamin Cramer, Mihai A Petrovici, Korbinian Schreiber, David Kappel, Johannes Schemmel, and Karlheinz Meier. Structural plasticity on an accelerated analog neuromorphic hardware system. *Neural Networks*, 133:11–20, 2021. URL <https://doi.org/10.1016/j.neunet.2020.09.024>.
- [62] Richard George, Giacomo Indiveri, and Stefano Vassanelli. Activity dependent structural plasticity in neuromorphic systems. In *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, 2017. URL <https://doi.org/10.1109/BIOCAS.2017.8325074>.
- [63] Arjen van Ooyen and Markus Butz-Ostendorf. *The rewiring brain: a computational approach to structural plasticity in the adult brain*. Academic Press, 2017. URL <https://doi.org/10.1016/B978-0-12-803784-3.00033-0>.
- [64] Javier Iglesias, Jan Eriksson, François Grize, Marco Tomassini, and Alessandro EP Villa. Dynamics of pruning in simulated large-scale spiking neural networks. *Biosystems*, 79(1-3):11–20, 2005. URL <https://doi.org/10.1016/j.biosystems.2004.09.016>.
- [65] Jeff Hawkins and Subutai Ahmad. Why neurons have thousands of synapses, a theory of sequence memory in neocortex. *Frontiers in Neural Circuits*, page 23, 2016. URL <https://doi.org/10.3389/fncir.2016.00023>.
- [66] Subhrajit Roy, Amitava Banerjee, and Arindam Basu. Liquid state machine with dendritically enhanced readout for low-power, neuromorphic vlsi implementations. *IEEE Transactions on Biomedical Circuits and Systems*, 8(5):681–695, 2014. URL <https://doi.org/10.1109/TBCAS.2014.2362969>.
- [67] Mark A Bourjaily and Paul Miller. Synaptic plasticity and connectivity requirements to produce stimulus-pair specific responses in recurrent networks of spiking neurons. *PLOS Computational Biology*, 7(2):e1001091, 2011. URL <https://doi.org/10.1371/journal.pcbi.1001091>.
- [68] Markus Butz, Ines D Steenbuck, and Arjen van Ooyen. Homeostatic structural plasticity increases the efficiency of small-world networks. *Frontiers in Synaptic Neuroscience*, 6:7, 2014. URL <https://doi.org/10.3389/fnsyn.2014.00007>.
- [69] Michael Fauth and Christian Tetzlaff. Opposing effects of neuronal activity on structural plasticity. *Frontiers in Neuroanatomy*, 10, 2016. URL <https://doi.org/10.3389/fnana.2016.00075>.
- [70] Sandra Diaz-Pier, Mikaël Naveau, Markus Butz-Ostendorf, and Abigail Morrison. Automatic generation of connectivity for large-scale neuronal network

- models through structural plasticity. *Frontiers in Neuroanatomy*, 10:57, 2016. URL <https://doi.org/10.3389/fnana.2016.00057>.
- [71] Christian Nowke, Sandra Diaz-Pier, Benjamin Weyers, Bernd Hentschel, Abigail Morrison, Torsten W Kuhlen, and Alexander Peyser. Toward rigorous parameterization of underconstrained neural network models through interactive visualization and steering of connectivity generation. *Frontiers in Neuroinformatics*, 12:32, 2018. URL <https://doi.org/10.3389/fninf.2018.00032>.
- [72] Donald Olding Hebb. *The Organisation of Behaviour: A Neuropsychological Theory*. Wiley, 1949. URL <https://doi.org/10.4324/9781410612403>.
- [73] Jeffrey C. Magee and Christine Grienberger. Synaptic plasticity forms and functions. *Annual Review of Neuroscience*, 43(1):95–117, 2020. URL <https://doi.org/10.1146/annurev-neuro-090919-022842>.
- [74] Abigail Morrison, Markus Diesmann, and Wulfram Gerstner. Phenomenological models of synaptic plasticity based on spike timing. *Biological Cybernetics*, 98(6):459–478, 2008. URL <https://doi.org/10.1007/s00422-008-0233-1>.
- [75] Shigeo Okabe, Hong-Duck Kim, Akiko Miwa, Toshihiko Kuriu, and Haruo Okado. Continual remodeling of postsynaptic density and its regulation by synaptic activity. *Nature Neuroscience*, 2(9):804–811, 1999.
- [76] Misha V. Tsodyks and Henry Markram. The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *Proceedings of the National Academy of Sciences*, 94:719–723, 1997. URL <https://doi.org/10.1073/pnas.94.2.719>.
- [77] Henry Markram, Yun Wang, and Misha Tsodyks. Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences*, 95(9):5323–5328, 1998. URL <https://doi.org/10.1073/pnas.95.9.5323>.
- [78] Anirudh Gupta, Yun Wang, and Henry Markram. Organizing principles for a diversity of GABAergic interneurons and synapses in the neocortex. *Science*, 287(5451):273–278, 2000. URL <https://doi.org/10.1126/science.287.5451.273>.
- [79] Robert S. Zucker and Wade G. Regehr. Short-term synaptic plasticity. *Annual Review of Physiology*, 64(1):355–405, 2002. URL <https://doi.org/10.1146/annurev.physiol.64.092501.114547>.

- [80] Henry Markram, Joachim Lübke, Michael Frotscher, Arnd Roth, and Bert Sakmann. Physiology and anatomy of synaptic connections between thick tufted pyramidal neurones in the developing rat neocortex. *Journal of Physiology*, 500(2):409–440, 1997. URL <https://doi.org/10.1113/jphysiol.1997.sp022031>.
- [81] Guo-qiang Bi and Mu-ming Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and post-synaptic cell type. *Journal of Neuroscience*, 18(24):10464–10472, 1998. URL <https://doi.org/10.1523/JNEUROSCI.18-24-10464.1998>.
- [82] Natalia Caporale and Yang Dan. Spike timing–dependent plasticity: a Hebbian learning rule. *Annual Reviews Neuroscience*, 31:25–46, 2008. URL <https://doi.org/10.1146/annurev.neuro.31.060407.125639>.
- [83] Henry Markram, Wulfram Gerstner, and Per Jesper Sjöström. Spike-timing-dependent plasticity: a comprehensive overview. *Frontiers in Synaptic Neuroscience*, 4:2, 2012. URL <https://doi.org/10.3389/fnsyn.2012.00002>.
- [84] Zuzanna Brzosko, Susanna B Mierau, and Ole Paulsen. Neuromodulation of spike-timing-dependent plasticity: past, present, and future. *Neuron*, 103(4):563–581, 2019. URL <https://doi.org/10.1016/j.neuron.2019.05.041>.
- [85] Yi Li, Yingpeng Zhong, Jinjian Zhang, Lei Xu, Qing Wang, Huajun Sun, Hao Tong, Xiaoming Cheng, and Xiangshui Miao. Activity-dependent synaptic plasticity of a chalcogenide electronic synapse for neuromorphic systems. *Scientific Reports*, 4(1):4906, 2014. URL <https://doi.org/10.1038/srep04906>.
- [86] Abigail Morrison, Ad Aertsen, and Markus Diesmann. Spike-timing-dependent plasticity in balanced random networks. *Neural Computation*, 19(6):1437–1467, 2007. URL <https://doi.org/10.1162/neco.2007.19.6.1437>.
- [87] Mark CW van Rossum, Guo Qiang Bi, and Gina G Turrigiano. Stable Hebbian learning from spike timing-dependent plasticity. *Journal of Neuroscience*, 20(23):8812–8821, 2000. URL <https://doi.org/10.1523/JNEUROSCI.20-23-08812.2000>.
- [88] Anthony N Burkitt, Hamish Meffin, and David B Grayden. Spike-timing-dependent plasticity: the relationship to rate-based learning for models with weight dynamics determined by a stable fixed point. *Neural Computation*, 16(5):885–940, 2004. URL <https://doi.org/10.1162/089976604773135041>.

- [89] Claudia Clopath and Wulfram Gerstner. Voltage and spike timing interact in STDP—a unified model. *Frontiers in Synaptic Neuroscience*, 2:25, 2010. URL <https://doi.org/10.3389/fnsyn.2010.00025>.
- [90] Claudia Clopath, Lars Busing, Eleni Vasilaki, and Wulfram Gerstner. Connectivity reflects coding: a model of voltage-based STDP with homeostasis. *Nature Neuroscience*, 13:344–352, 2010. URL <https://doi.org/10.1038/nn.2479>.
- [91] Anacleto Ngezahayo, Melitta Schachner, and Alain Artola. Synaptic activity modulates the induction of bidirectional synaptic changes in adult mouse hippocampus. *Journal of Neuroscience*, 20(7):2451–2458, 2000. URL <https://doi.org/10.1523/JNEUROSCI.20-07-02451.2000>.
- [92] Per Jesper Sjöström, Gina G Turrigiano, and Sacha B Nelson. Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron*, 32(6):1149–1164, 2001. URL [https://doi.org/10.1016/S0896-6273\(01\)00542-6](https://doi.org/10.1016/S0896-6273(01)00542-6).
- [93] Robert Urbanczik and Walter Senn. Learning by the dendritic prediction of somatic spiking. *Neuron*, 81(3):521–528, 2014. URL <https://doi.org/10.1016/j.neuron.2013.11.030>.
- [94] A. Artola, S Bröcher, and W Singer. Different voltage dependent thresholds for inducing long-term depression and long-term potentiation in slices of rat visual cortex. *Nature*, 347:69–72, 1990. URL <https://doi.org/10.1038/347069a0>.
- [95] Verena Pawlak, Jeffery R Wickens, Alfredo Kirkwood, and Jason ND Kerr. Timing is not everything: neuromodulation opens the STDP gate. *Frontiers in Synaptic Neuroscience*, 2:146, 2010. URL <https://doi.org/10.3389/fnsyn.2010.00146>.
- [96] Florentin Wörgötter and Bernd Porr. Temporal sequence learning, prediction, and control: a review of different models and their relation to biological mechanisms. *Neural Computation*, 17(2):245–319, 2005. URL <https://doi.org/10.1162/0899766053011555>.
- [97] Jonas Stapmanns, Jan Hahne, Moritz Helias, Matthias Bolten, Markus Diesmann, and David Dahmen. Event-based update of synapses in voltage-based learning rules. *Frontiers in Neuroinformatics*, 15, 2021. URL <https://doi.org/10.3389/fninf.2021.609147>.
- [98] Abigail Morrison, Carsten Mehring, Theo Geisel, AD Aertsen, and Markus Diesmann. Advancing the boundaries of high-connectivity network simulation

- p>with distributed computing.
- Neural Computation*
- , 17(8):1776–1801, 2005. URL
- <https://doi.org/10.1162/0899766054026648>
- .
- [99] Philipp Weidel, Renato Duarte, and Abigail Morrison. Unsupervised learning and clustered connectivity enhance reinforcement learning in spiking neural networks. *Frontiers in Computational Neuroscience*, 15:18, 2021. URL <https://doi.org/10.3389/fncom.2021.543872>.
 - [100] Tim P Vogels, Henning Sprekeler, Friedemann Zenke, Claudia Clopath, and Wulfram Gerstner. Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science*, 334(6062):1569–1573, 2011. URL <https://doi.org/10.1126/science.1211095>.
 - [101] Jakob Jordan, Maximilian Schmidt, Walter Senn, and Mihai A Petrovici. Evolving interpretable plasticity for spiking networks. *eLife*, 10:e66273, 2021. URL <https://doi.org/10.7554/eLife.66273>.
 - [102] Adam B. Barrett, Guy O. Billings, Richard G. M. Morris, and Mark C. W. van Rossum. State based model of long-term potentiation and synaptic tagging and capture. *PLOS Computational Biology*, 5(1):e1000259, 2009. URL <https://doi.org/10.1371/journal.pcbi.1000259>.
 - [103] Gianluigi Mongillo, Omri Barak, and Misha Tsodyks. Synaptic theory of working memory. *Science*, 319(5869):1543–1546, 2008. URL <https://doi.org/10.1126/science.1150769>.
 - [104] Christof Koch and Gilles Laurent. Complexity and the nervous system. *Science*, 284(5411):96–98, 1999. URL <https://doi.org/10.1126/science.284.5411.96>.
 - [105] Renato Duarte and Abigail Morrison. Leveraging heterogeneity for neural computation with fading memory in layer 2/3 cortical microcircuits. *PLOS Computational Biology*, 15(4):e1006781, 2019. URL <https://doi.org/10.1371/journal.pcbi.1006781>.
 - [106] Vladimir Parpura and Robert Zorec. Gliotransmission: exocytotic release from astrocytes. *Brain Research Reviews*, 63(1-2):83–92, 2010. URL <https://doi.org/10.1016/j.brainresrev.2009.11.008>.
 - [107] P. Girard, J. M. Hupé, and J. Bullier. Feedforward and feedback connections between areas V1 and V2 of the monkey have similar rapid conduction velocities. *Journal of Neurophysiology*, 85(3):1328–1331, 2001. URL <https://doi.org/10.1152/jn.2001.85.3.1328>.

- [108] Daniel Liewald, Robert Miller, Nikos Logothetis, Hans-Joachim Wagner, and Almut Schüz. Distribution of axon diameters in cortical white matter: an electron-microscopic study on three human brains and a macaque. *Biological Cybernetics*, 108(5):541–557, 2014. URL <https://doi.org/10.1007/s00422-014-0626-2>.
- [109] Lyle Muller, Frédéric Chavane, John Reynolds, and Terrence J. Sejnowski. Cortical travelling waves: mechanisms and computational principles. *Nature Reviews Neuroscience*, 19(5):255–268, 2018. URL <https://doi.org/10.1038/nrn.2018.20>.
- [110] Sen Song, Per Sjöström, Markus Reigl, Sacha Nelson, and Dmitri Chklovskii. Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLOS Computational Biology*, 3(3):e68, 2005. URL <https://doi.org/10.1371/journal.pbio.0030068>.
- [111] Sandrine Lefort, Christian Tamm, J.-C. Floyd Sarria, and Carl C. H. Petersen. The excitatory neuronal network of the C2 barrel column in mouse primary somatosensory cortex. *Neuron*, 61:301–316, 2009. URL <https://doi.org/10.1016/j.neuron.2008.12.020>.
- [112] Alexei A. Koulakov, Tomas Hromádka, and Anthony M. Zador. Correlated connectivity and the distribution of firing rates in the neocortex. *Journal of Neuroscience*, 29(12):3685–3694, 2009. URL <https://doi.org/10.1523/JNEUROSCI.4500-08.2009>.
- [113] M Avermann, C Tamm, C Mateo, W Gerstner, and Carl CH Petersen. Microcircuits of excitatory and inhibitory neurons in layer 2/3 of mouse barrel cortex. *Journal of Neurophysiology*, 107(11):3116–3134, 2012. URL <https://doi.org/10.1152/jn.00917.2011>.
- [114] Yuji Ikegaya, Takuya Sasaki, Daisuke Ishikawa, Naoko Honma, Kentaro Tao, Naoya Takahashi, Genki Minamisawa, Sakiko Ujita, and Norio Matsuki. Interpyramid spike transmission stabilizes the sparseness of recurrent network activity. *Cerebral Cortex*, 23(2):293–304, 2013. URL <https://doi.org/10.1093/cercor/bhs006>.
- [115] A. Kuhn, A. Aertsen, and S. Rotter. Neuronal integration of synaptic input in the fluctuation-driven regime. *Journal of Neuroscience*, 24(10):2345–2356, 2004. URL <https://doi.org/10.1523/JNEUROSCI.3349-03.2004>.
- [116] Alex Roxin, Nicolas Brunel, David Hansel, Gianluigi Mongillo, and Carl van Vreeswijk. On the distribution of firing rates in networks of cortical neurons.

- Journal of Neuroscience*, 31(45):16217–16226, 2011. URL <https://doi.org/10.1523/jneurosci.1677-11.2011>.
- [117] Nicolas Brunel and Vincent Hakim. Fast global oscillations in networks of integrate-and-fire neurons with low firing rates. *Neural Computation*, 11(7):1621–1671, 1999. URL <https://doi.org/10.1162/089976699300016179>.
- [118] B. M. Kampa, J. J. Letzkus, and G. J. Stuart. Dendritic mechanisms controlling spike-timing-dependent synaptic plasticity. *Trends in Neuroscience*, 30(9):456–463, 2007. URL <https://doi.org/10.1016/j.tins.2007.06.010>.
- [119] Dirk Feldmeyer, Veronica Egger, Joachim Lübke, and Bert Sakmann. Reliable synaptic connections between pairs of excitatory layer 4 neurones within a single ‘barrel’ of developing rat somatosensory cortex. *Journal of Physiology*, 521(1):169–190, 1999. URL <https://doi.org/10.1111/j.1469-7793.1999.00169.x>.
- [120] Dirk Feldmeyer, Joachim Lübke, R Angus Silver, and Bert Sakmann. Synaptic connections between layer 4 spiny neurone-layer 2/3 pyramidal cell pairs in juvenile rat barrel cortex: physiology and anatomy of interlaminar signalling within a cortical column. *Journal of Physiology*, 538(3):803–822, 2002. URL <https://doi.org/10.1113/jphysiol.2001.012959>.
- [121] Dirk Feldmeyer, Joachim Lübke, and Bert Sakmann. Efficacy and connectivity of intracolumnar pairs of layer 2/3 pyramidal cells in the barrel cortex of juvenile rats. *Journal of Physiology*, 575:583–602, 2006. URL <https://doi.org/10.1113/jphysiol.2006.105106>.
- [122] A. Stepanyants, J. Hirsch, L. M. Martinez, Z. F. Kisvarday, A. S. Ferecsko, and D. B. Chklovskii. Local potential connectivity in cat primary visual cortex. *Cerebral Cortex*, 18(1):13–28, 2008. URL <https://doi.org/10.1093/cercor/bhm027>.
- [123] Alex Roxin. The role of degree distribution in shaping the dynamics in networks of sparsely connected spiking neurons. *Frontiers in Computational Neuroscience*, 5(8), 2011. URL <https://doi.org/10.3389/fncom.2011.00008>.
- [124] C. van Vreeswijk and H. Sompolinsky. Chaotic balanced state in a model of cortical circuits. *Neural Computation*, 10(6):1321–1371, 1998. URL <https://doi.org/10.1162/089976698300017214>.
- [125] M. Tsodyks, I. Mitkov, and H. Sompolinsky. Pattern of synchrony in inhomogeneous networks of oscillators with pulse interactions. *Physical Review Letters*, 71(8), 1993. URL <https://doi.org/10.1103/PhysRevLett.71.1280>.

- [126] David Golomb and John Rinzel. Dynamics of globally coupled inhibitory neurons with heterogeneity. *Physical Review Letters*, 48(6):4810–4814, 1993. URL <https://doi.org/10.1103/PhysRevE.48.4810>.
- [127] L. Neltner, D. Hansel, G. Mato, and C. Meunier. Synchrony in heterogeneous networks of spiking neurons. *Neural Computation*, 12(7):1607–1641, 2000. URL <https://doi.org/10.1162/089976600300015286>.
- [128] Michael Denker, Marc Timme, Markus Diesmann, Fred Wolf, and Theo Geisel. Breaking synchrony by heterogeneity in complex networks. *Physical Review Letters*, 92(7):074103–1–074103–4, 2004. URL <https://doi.org/10.1103/PhysRevLett.92.074103>.
- [129] J. F. Mejias and A. Longtin. Optimal heterogeneity for coding in spiking neural networks. *Physical Review Letters*, 108, 2012. URL <https://doi.org/10.1103/PhysRevLett.108.228102>.
- [130] Thomas Pfeil, Jakob Jordan, Tom Tetzlaff, Andreas Grübl, Johannes Schemmel, Markus Diesmann, and Karlheinz Meier. Effect of heterogeneity on decorrelation mechanisms in spiking neural networks: A neuromorphic-hardware study. *Physical Review X*, 6:021023, 2016. URL <https://doi.org/10.1103/PhysRevX.6.021023>.
- [131] Nigel G. Stocks. Suprathreshold stochastic resonance in multilevel threshold systems. *Physical Review Letters*, 84(11):2310, 2000. URL <https://doi.org/10.1103/PhysRevLett.84.2310>.
- [132] M Shamir and H Sompolinsky. Implications of neuronal diversity on population coding. *Neural Computation*, 18(8):1951–1986, 2006. URL <https://doi.org/10.1162/neco.2006.18.8.1951>.
- [133] Mircea I. Chelaru and Valentin Dragoi. Efficient coding in heterogeneous neuronal populations. *Proceedings of the National Academy of Sciences*, 105(42), 2008. URL <https://doi.org/10.1073/pnas.0807744105>.
- [134] Leslie C Osborne, Stephanie E Palmer, Stephen G Lisberger, and William Bialek. The neural basis for combinatorial coding in a cortical population response. *Journal of Neuroscience*, 28(50):13522–13531, 2008. URL <https://doi.org/10.1523/JNEUROSCI.4390-08.2008>.
- [135] Krishnan Padmanabhan and Nathaniel N Urban. Intrinsic biophysical diversity decorrelates neuronal firing while increasing information content. *Nature Neuroscience*, 13(10):1276–1282, 2010. URL <https://doi.org/10.1038/nn.2630>.

- [136] Gary Marsat and Leonard Maler. Neural heterogeneity and efficient population codes for communication signals. *Journal of Neurophysiology*, 104(5):2543–2555, 2010. URL <https://doi.org/10.1152/jn.00256.2010>.
- [137] Lars A Holmstrom, Lonneke BM Eeuwes, Patrick D Roberts, and Christine V Portfors. Efficient encoding of vocalizations in the auditory midbrain. *Journal of Neuroscience*, 30(3):802–819, 2010. URL <https://doi.org/10.1523/JNEUROSCI.1964-09.2010>.
- [138] Man Yi Yim, Ad Aertsen, and Stefan Rotter. Impact of intrinsic biophysical diversity on the activity of spiking neurons. *Physical Review E*, 87, 2013. URL <https://doi.org/10.1103/PhysRevE.87.032710>.
- [139] Johannes Lengler, Florian Jug, and Angelika Steger. Reliable neuronal systems: The importance of heterogeneity. *PLOS One*, 2013. URL <https://doi.org/10.1371/journal.pone.0080694>.
- [140] Jorge F Mejias and André Longtin. Differential effects of excitatory and inhibitory heterogeneity on the gain and asynchronous state of sparse cortical networks. *Frontiers in Computational Neuroscience*, 8, 2014. URL <https://doi.org/10.3389/fncom.2014.00107>.
- [141] Demian Battaglia, Anastassios Karagiannis, Thierry Gallopin, Harold W Gutch, and Bruno Cauli. Beyond the frontiers of neuronal types. *Frontiers in Neural Circuits*, 7:13, 2013. URL <https://doi.org/10.3389/fncir.2013.00013>.
- [142] György Buzsáki and Kenji Mizuseki. The log-dynamic brain: how skewed distributions affect network operations. *Nature Reviews Neuroscience*, 15(4):264–278, 2014. URL <https://doi.org/10.1038/nrn3687>.
- [143] PA Robinson, Xiao Gao, and Yinuo Han. Relationships between lognormal distributions of neural properties, activity, criticality, and connectivity. *Biological Cybernetics*, 115(2):121–130, 2021. URL <https://doi.org/10.1007/s00422-021-00871-z>.
- [144] Aitor Morales-Gregorio, Alexander van Meegen, and Sacha Jennifer van Albada. Ubiquitous lognormal distribution of neuron densities across mammalian cerebral cortex. *bioRxiv*, 2022. URL <https://doi.org/10.1101/2022.03.17.480842>.
- [145] Astrid A Prinz, Dirk Bucher, and Eve Marder. Similar network activity from disparate circuit parameters. *Nature Neuroscience*, 7:1345–1352, 2004. URL <https://doi.org/10.1038/nn1352>.

- [146] Pablo Achard and Erik De Schutter. Complex parameter landscape for a complex neuron model. *PLOS Computational Biology*, 2(7):e94, 2006. URL <https://doi.org/10.1371/journal.pcbi.0020094>.
- [147] Jyotika Bahuguna, Tom Tetzlaff, Arvind Kumar, Jeanette Hellgren Kotaleski, and Abigail Morrison. Homologous basal ganglia network models in physiological and Parkinsonian conditions. *Frontiers in Computational Neuroscience*, 11:79, 2017. URL <https://doi.org/10.3389/fncom.2017.00079>.
- [148] L. F. Abbott and S. B. Nelson. Synaptic plasticity: taming the beast. *Nature Neuroscience*, 3:1178–1183, 2000. URL <https://doi.org/10.1038/81453>.
- [149] Christian Tetzlaff, Christoph Kolodziejewski, Marc Timme, and Florentin Wörgötter. Synaptic scaling in combination with many generic plasticity mechanisms stabilizes circuit connectivity. *Frontiers in Computational Neuroscience*, 5:47, 2011. URL <https://doi.org/10.3389/fncom.2011.00047>.
- [150] Chris Eliasmith and Charles H Anderson. *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press, 2002. URL <https://mitpress.mit.edu/9780262050715/>.
- [151] Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature Communications*, page 3625, 2020. URL <https://doi.org/10.1038/s41467-020-17236-y>.
- [152] Steve Furber. Large-scale neuromorphic computing systems. *Journal of Neural Engineering*, 13(5):051001, 2016. URL <https://doi.org/10.1088/1741-2560/13/5/051001>.
- [153] Timo Wunderlich, Akos F. Kungl, Eric Müller, Andreas Hartel, Yannik Stradmann, Syed Ahmed Aamir, Andreas Grübl, Arthur Heimbrecht, Korbinian Schreiber, David Stöckel, Christian Pehle, Sebastian Billaudelle, Gerd Kiene, Christian Mauch, Johannes Schemmel, Karlheinz Meier, and Mihai A. Petrovici. Demonstrating advantages of neuromorphic computation: A pilot study. *Frontiers in Neuroscience*, 13, 2019. URL <https://doi.org/10.3389/fnins.2019.00260>.
- [154] Yury V Zaytsev and Abigail Morrison. Increasing quality and managing complexity in neuroinformatics software development with continuous integration. *Frontiers in Neuroinformatics*, 6:31, 2013. URL <https://doi.org/10.3389/fninf.2012.00031>.

- [155] Hans E Plesser. Reproducibility vs. replicability: a brief history of a confused terminology. *Frontiers in Neuroinformatics*, 11:76, 2018. URL <https://doi.org/10.3389/fninf.2017.00076>.
- [156] Steven N. Goodman, Daniele Fanelli, and John P. A. Ioannidis. What does research reproducibility mean? *Science Translational Medicine*, 8(341), 2016. URL <https://doi.org/10.1126/scitranslmed.aaf5027>.
- [157] S. J. van Albada, S. Kunkel, A. Morrison, and M. Diesmann. Integrating brain structure and dynamics on supercomputers. In *Brain-Inspired Computing*, pages 22–32, 2014. URL https://doi.org/10.1007/978-3-319-12084-3_3.
- [158] Anno C Kurth, Johanna Senk, Dennis Terhorst, Justin Finnerty, and Markus Diesmann. Sub-realtime simulation of a neuronal network of natural density. *Neuromorphic Computing and Engineering*, 2(2):021001, 2022. URL <https://doi.org/10.1088/2634-4386/ac55fc>.
- [159] Alexander Hanuschkin, Susanne Kunkel, Moritz Helias, Abigail Morrison, and Markus Diesmann. A general and efficient method for incorporating precise spike times in globally time-driven simulations. *Frontiers in Neuroinformatics*, 4, 2010. URL <https://doi.org/10.3389/fninf.2010.00113>.
- [160] J. N. Teramae, Y. Tsubo, and T. Fukai. Optimal spike-based communication in excitable networks with strong-sparse and weak-dense links. *Scientific Reports*, 2:485, 2012. URL <https://doi.org/10.1038/srep00485>.
- [161] SL Malkin, K Kh Kim, DB Tikhonov, and AV Zaitsev. Properties of spontaneous and miniature excitatory postsynaptic currents in neurons of the rat prefrontal cortex. *Journal of Evolutionary Biochemistry and Physiology*, 50(6):506–514, 2014. URL <https://doi.org/10.1134/S0022093014060052>.
- [162] Stefan Dasbach, Tom Tetzlaff, Markus Diesmann, and Johanna Senk. Dynamical characteristics of recurrent neuronal networks are robust against low synaptic weight resolution. *Frontiers in Neuroscience*, 15, 2021. URL <https://doi.org/10.3389/fnins.2021.757790>.
- [163] Thomas Pfeil, Tobias C Potjans, Sven Schrader, Wiebke Potjans, Johannes Schemmel, Markus Diesmann, and Karlheinz Meier. Is a 4-bit synaptic weight resolution enough?—constraints on enabling spike-timing dependent plasticity in neuromorphic hardware. *Frontiers in Neuroscience*, 6:90, 2012. URL <https://doi.org/10.3389/fnins.2012.00090>.

- [164] Eilen Nordlie, Marc-Oliver Gewaltig, and Hans Ekkehard Plesser. Towards reproducible descriptions of neuronal network models. *PLOS Computational Biology*, 5(8):1–18, 2009. URL <https://doi.org/10.1371/journal.pcbi.1000456>.
- [165] Inga Blundell, Romain Brette, Thomas A Cleland, Thomas G Close, Daniel Coca, Andrew P Davison, Sandra Diaz-Pier, Carlos Fernandez Musoles, Padraig Gleeson, Dan FM Goodman, et al. Code generation in computational neuroscience: a review of tools and techniques. *Frontiers in Neuroinformatics*, 12:68, 2018. URL <https://doi.org/10.3389/fninf.2018.00068>.
- [166] Robert A. McDougal, Thomas M. Morse, Ted Carnevale, Luis Marengo, Rixin Wang, Michele Migliore, Perry L. Miller, Gordon M. Shepherd, and Michael L. Hines. Twenty years of ModelDB and beyond: building essential modeling tools for the future of neuroscience. *Journal of Computational Neuroscience*, 42(1): 1–10, 2016. URL <https://doi.org/10.1007/s10827-016-0623-7>.
- [167] Padraig Gleeson, Matteo Cantarelli, Boris Marin, Adrian Quintana, Matt Earnshaw, Sadra Sadeh, Eugenio Piasini, Justas Birgiolas, Robert C Cannon, N Alex Cayco-Gajic, et al. Open Source Brain: a collaborative resource for visualizing, analyzing, simulating, and developing standardized models of neurons and circuits. *Neuron*, 103(3):395–411, 2019. URL <https://doi.org/10.1016/j.neuron.2019.05.019>.
- [168] Anthony Holtmaat, Tobias Bonhoeffer, David K Chow, Jyoti Chuckowree, Vincenzo De Paola, Sonja B Hofer, Mark Hübener, Tara Keck, Graham Knott, Wei-Chung A Lee, et al. Long-term, high-resolution imaging in the mouse neocortex through a chronic cranial window. *Nature Protocols*, 4(8):1128–1144, 2009. URL <https://doi.org/10.1038/nprot.2009.89>.
- [169] Sean Foxley, Vandana Sampathkumar, Vincent De Andrade, Scott Trinkle, Anastasia Sorokina, Katrina Norwood, Patrick La Riviere, and Narayanan Kasthuri. Multi-modal imaging of a single mouse brain over five orders of magnitude of resolution. *NeuroImage*, 238:118250, 2021. URL <https://doi.org/10.1016/j.neuroimage.2021.118250>.
- [170] Chris Chatfield. Model uncertainty, data mining and statistical inference. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 158(3):419, 1995. URL <https://doi.org/10.2307/2983440>.

9 Links to the Programs

1. https://nest-simulator.readthedocs.io/en/latest/tutorials/pynest_tutorial/part_3_connecting_networks_with_synapses.html
2. https://github.com/nest/nestml/blob/master/models/neurons/wb_cond_multisyn.nestml
3. https://www.nest-simulator.org/py_sample/structural_plasticity/
4. https://nest-simulator.readthedocs.io/en/latest/auto_examples/tsodyks_depressing.html
5. https://nest-simulator.readthedocs.io/en/latest/auto_examples/tsodyks_facilitating.html
6. https://nestml.readthedocs.io/en/latest/models_library/stdp.html
7. https://nestml.readthedocs.io/en/latest/nestml_language/synapses_in_nestml.html
8. https://nest-simulator.readthedocs.io/en/latest/guides/weight_normalization.html
9. https://nest-extension-module.readthedocs.io/en/latest/extension_modules.html