

# Noise as a Resource for Computation and Learning in Networks of Spiking Neurons

*This paper discusses biologically inspired machine learning methods based on theories about how the brain exploits noise to carry out computations, such as probabilistic inference through sampling.*

By WOLFGANG MAASS

**ABSTRACT** | We are used to viewing noise as a nuisance in computing systems. This is a pity, since noise will be abundantly available in energy-efficient future nanoscale devices and circuits. I propose here to learn from the way the brain deals with noise, and apparently even benefits from it. Recent theoretical results have provided insight into how this can be achieved: how noise enables networks of spiking neurons to carry out probabilistic inference through sampling and also enables creative problem solving. In addition, noise supports the self-organization of networks of spiking neurons, and learning from rewards. I will sketch here the main ideas and some consequences of these results. I will also describe why these results are paving the way for a qualitative jump in the computational capability and learning performance of neuromorphic networks of spiking neurons with noise, and for other future computing systems that are able to treat noise as a resource.

**KEYWORDS** | Computational power; neural networks; neuromorphic hardware; noise; self-organization; spiking neurons; stochastic computing

## I. INTRODUCTION

Quite a number of algorithms and architectures have been proposed for computations with spiking neurons. Some of them have also been implemented in dedicated “neuro-

morphic” hardware, i.e., in analog/digital very large-scale integration (VLSI) systems that mimic neurobiological architectures present in the nervous system (see, e.g., [1]–[3]). But virtually all of them ([4] for an exception) treat spiking neurons as deterministic computing elements, in spite of the fact that biological neurons and synapses are not deterministic. If noise is considered at all, it is typically handled as a nuisance and eliminated through duplications and averaging, which drastically reduces the computational efficiency of networks of spiking neurons. A quite intriguing idea is that noise and trial-to-trial variability are present in biological networks of neurons not only as a consequence of inherently stochastic processes on the molecular level, but also because they are salient components of the computational strategy of the brain.

One major source of stochasticity in networks of neurons in the brain is the unreliability of synapses. A spike of a presynaptic neuron causes a release of a vesicle filled with neurotransmitters at a synaptic release site with low probability (around 0.1 according to [5]). Two synaptically connected pyramidal cells in the cortex are typically connected via five to ten of such release sites, but this still causes a highly unreliable synaptic transmission. In addition, synaptic vesicles are also released without a presynaptic spike, which contributes an additional source of noise. Synaptic release is particularly intriguing as a noise source in networks of neurons, since it is quite plausible that nature could have produced highly reliable synapses if needed. Furthermore, this noise source can apparently be modified through synaptic plasticity [6]. Other sources of noise in biological neurons, such as stochastic openings and closings of membrane channels, are due to the inherent stochasticity of processes on the molecular scale, and may be difficult to eliminate. Nice

Manuscript received November 18, 2013; revised January 24, 2014 and February 28, 2014; accepted March 6, 2014. Date of publication April 14, 2014; date of current version April 28, 2014. This work was supported in part by the European Union project # FP7-604102 (Human Brain Project).

The author is with the Institute for Theoretical Computer Science, Graz University of Technology, Graz 8010, Austria (e-mail: maass@igi.tugraz.at; <http://www.igi.tugraz.at/maass/>).

Digital Object Identifier: 10.1109/JPROC.2014.2310593

0018-9219 © 2014 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See [http://www.ieee.org/publications\\_standards/publications/rights/index.html](http://www.ieee.org/publications_standards/publications/rights/index.html) for more information.

reviews of noise sources in biological neurons are given in [7]–[11].

It has already been proposed a while ago that noise is beneficial for the detection of weak signals by neurons (“stochastic resonance”; see the review in [10]). In [10], also the term “stochastic facilitation” was coined to describe scenarios where specific computational goals are better achieved in the presence of noise, and various neurobiological experiments were proposed for testing such more general advantageous role in neural systems. But until very recently [12]–[16] concrete *computational* benefits of noise in networks of spiking neurons have apparently not been addressed in theoretical neuroscience research.

That noise can be a valuable computational resource is well known in computer science (see, e.g., [17] and [18]). Many practically important computations on digital computers employ so-called random number generators, i.e., algorithmic procedures that create pseudorandom sequences of bits that share important features with true random bits (that can, for example, be generated by coin tosses). Very recently, it has also been shown that sufficiently complex deterministic networks of spiking neurons can generate randomly looking spiking activity [19]–[22].

The value of noise as a resource for neural computation had already been addressed several decades ago in the context of artificial neural networks: Boltzmann machines [23]–[25] are networks of simple stochastic units with binary outputs that are of interest both from the perspective of theoretical neuroscience and from the more general perspective of massively parallel stochastic computation. They switch their output bit with a probability that depends on the current weighted sum of their input bits: when unit  $i$  is considered for switching, it generates the output

$$x_i = 1 \text{ with probability } \sigma\left(\frac{1}{T}\left(\sum_j w_{ij}x_j + b_i\right)\right) \quad (1)$$

else  $x_i = 0$ . Parameters  $w_{ij}$  denote here the synaptic weights between units  $i$  and  $j$  (with the requirement of symmetry:  $w_{ij} = w_{ji}$ ),  $b_i$  denotes a bias for unit  $i$ ,  $T$  (“temperature”) determines the level of noise in the units, and  $\sigma(x) = 1/(1 + e^{-x})$  denotes the common sigmoid function. The value of  $x_i$  stays fixed until unit  $i$  is considered next time for switching. The state of a Boltzmann machine at any time  $t$  is the binary vector  $\langle x_1, \dots, x_m \rangle$  of the current output values of its  $m$  units, according to some global schedule (which can be stochastic). Note that the units of a Boltzmann machine are not autonomously active (like spiking neurons), but require a global schedule.

Boltzmann machines can, on the one hand, be viewed as stochastic versions of the deterministic Hopfield

network [26] that emerges from the previously given switching rule (1) if one lets temperature  $T$  go to 0. Any nonzero temperature enables state  $\langle x_1, \dots, x_m \rangle$  of a Boltzmann machines to escape from local minima of its so-called *energy function*

$$E(\langle x_1, \dots, x_m \rangle) = -\sum_{i < j} w_{ij}x_i x_j - \sum_i b_i x_i \quad (2)$$

whereas a deterministic Hopfield network gets stuck there. Therefore, Boltzmann machines can be viewed from a more abstract computational perspective as architectures for solving large constraint satisfaction problems (for many variables  $x_1, \dots, x_m$ ) through massively parallel stochastic search. A weight  $w_{ij}$  can be viewed in this context as a (soft) constraint: A positive value of  $w_{ij}$  encodes a preference for network states  $\langle x_1, \dots, x_m \rangle$  with  $x_i = x_j = 1$  (and in combination with suitable values for  $b_i$  and  $b_j$ , one can also encode a more general preference for  $x_i = x_j$ ), since this minimizes their contribution  $-w_{ij}x_i x_j$  to the energy function (2). Such preference for correlated values of  $x_i$  and  $x_j$  is, for example, meaningful if  $x_i$  represents binary pixel values of an image that consists of black and white areas. In this case, the stochastic dynamics of the Boltzmann machine eliminates local pixel noise by moving the network to states with lower energy. Finally, Boltzmann machines are of interest from the more general perspective of stochastic dynamical systems and machine learning. They are special cases of Markov chains, i.e., of the arguably simplest models for stochastic systems with discrete time and finite sets of states. Markov chains move at each discrete time step from their current state  $a$  to another state  $b$  (where  $b = a$  is in general allowed) according to some given time-invariant conditional probability  $p(b|a)$ . Boltzmann machines are special cases of Markov chains: they are in addition reversible. This means that transitions between any two network states  $a$  and  $b$  occur with the same probability in either direction (note that the probability of a transition from  $a$  to  $b$  is given by *product*  $p(a) \times p(b|a)$  of probability  $p(a)$  of being in state  $a$  and choosing from  $a$  a transition to  $b$ ).

The dynamics of Boltzmann machines can be viewed as a special case of Markov chain Monte Carlo (MCMC) sampling, which is an important approach to carry out probabilistic inference for multivariable probability distributions by drawing and analyzing samples from them [27]. More precisely, the dynamics of Boltzmann machines is a special case of Gibbs sampling. Gibbs sampling [28] is a very useful approximation method for generating samples  $\langle x_1, \dots, x_m \rangle$  from a given joint distribution  $p$  over many variables. It does this through repeated local stochastic drawings of values of random variables (RVs)  $x_i$  according to the conditional probability of  $x_i$  under  $p$ , given the current values of the other RVs. A Boltzmann machine

generates through its stochastic local switching according to the previously specified conditional probability (1) after some “burn-in time” (during which its dependence on the initial state declines) samples  $\langle x_1, \dots, x_m \rangle$  from the multivariable joint distribution

$$p_B(\langle x_1, \dots, x_m \rangle) = \frac{1}{Z} e^{-E(\langle x_1, \dots, x_m \rangle)/T} \quad (3)$$

which is its unique stationary distribution of network states to which it converges from any initial state. This stationary distribution is a second-order distribution (a “Boltzmann distribution”) that can be characterized through the synaptic weights  $w_{ij}$  and biases  $b_i$  of its units ( $Z$  is a normalization factor). It assigns the highest probabilities to states with the lowest energy  $E(\langle x_1, \dots, x_m \rangle)$  according to (2). Hence, for computational applications, parameters  $w_{ij}$  and  $b_i$  are usually chosen (or learned) so that low-energy states correspond to desirable global solutions of an underlying constraint satisfaction problem.

Boltzmann machines are among the computationally most powerful types of artificial neural networks (see, e.g., [25]), and are known to achieve a performance in some perceptual tasks that rivals that of the best machine learning approaches; see, e.g., the results on deep learning in hierarchical Boltzmann machines [29]. The reason is that their inherent stochasticity creates direct links to powerful methods in statistical learning theory; in particular, it supports the implementation of generative models for learning. In fact, every distribution with at most second-order dependencies is a Boltzmann distribution.

Spiking neurons are in principle quite different from the units of a Boltzmann machine, and they had been introduced as more adequate models for biological neurons. In contrast to the units of Boltzmann machines or other artificial neural networks, the output of a biological neuron is not a discrete or analog number, but rather a brief increase in the membrane potential at its soma, called action potential or spike. Each spike is transmitted through branches of its axon via synaptic connections to other (“postsynaptic”) neurons, where it causes excitatory or inhibitory postsynaptic potentials (EPSPs or IPSPs). As described above, this transmission process is quite noisy. But for the sake of mathematical simplicity (and lack of precise data), this noise is usually projected into the spike generation process of the presynaptic neuron. In this way, one arrives at the following model for a stochastically firing neuron (brief: stochastic neuron) from [30].

A stochastic neuron  $\nu_i$  has at any time point  $t$  continuously an instantaneous firing probability

$$\rho_i(t) = \frac{1}{\tau} \exp(u_i(t)) \quad (4)$$

where  $u_i(t) = \sum_j w_{ij} \tilde{x}_j(t) + b_i$  models the membrane potential at its soma at time  $t$  and  $w_{ij} \tilde{x}_j(t)$  models the contribution of spikes from the presynaptic neuron  $\nu_j$  to this membrane potential. Thus,  $w_{ij} \tilde{x}_j(t)$  is a superposition of postsynaptic potentials (positive if  $\nu_j$  is an excitatory neuron; negative if  $\nu_j$  is an inhibitory neuron) of some length  $\tau$  (20 ms is a typical value) that are caused by spikes of neuron  $\nu_j$ . This simple phenomenological model can be fitted quite well to experimental data [30].

One can characterize the current state  $a$  of a network of  $m$  stochastically firing neurons at any time  $t$  by a binary vector  $\langle x_1, \dots, x_m \rangle$ , similarly as for a Boltzmann machine, where  $x_i = 1$  indicates now that neuron  $i$  has fired within the time interval  $[t - \tau, t]$ . But note that this binary vector  $a$  is *not* a Markov state of the underlying Markov chain, since that would require that the future firing activity of the network be stochastically independent from its activity before time  $t - \tau$ , given state  $a$ . A Markov state requires records of precise spike times.

From the perspective of these binary states, spiking neurons appear to be not so different from the stochastic computational units of Boltzmann machines. But surprisingly, little effort was made until very recently [12] to examine the portability of algorithms and architectures from Boltzmann machines to networks of spiking neurons with noise. One reason for that was perhaps that none of the two possible output values 1, 0 of a unit in a Boltzmann machine plays any special role: they are completely symmetric. In contrast, a spike is typically rather short (1–2 ms), and is, on average, followed by a much larger interspike interval of one or several hundred milliseconds. This asymmetry is essential for energy-efficient computation, since spikes (and their propagation to other neurons) consume substantially more energy than the output signal “no spike.”

But also the elegant theory of Boltzmann machines cannot directly be applied to networks of spiking neurons. In particular, two important components of the theory of Boltzmann machines get lost. One component that gets lost is the link between the dynamics of the network and Gibbs sampling. The reason is that the Markov chain that describes the stochastic dynamics of a Boltzmann machine is reversible (which is required for Gibbs sampling).

One can also model a network of stochastically firing neurons as a Markov chain (for a suitable notion of a Markov state, see Section II-A). But this Markov chain is nonreversible. This nonreversibility arises from the fact that a spike has, in general, an impact on the network state (e.g., through postsynaptic potentials) that transcends the duration of the spike itself, and is, in general, not reversible. Another important feature of Boltzmann machines that gets lost in a transition to spiking neurons is the possibility to directly relate the stationary distribution  $p_B$  of network states  $\langle x_1, \dots, x_m \rangle$  of a Boltzmann machine  $B$  to its parameters according to a simple formula such as (3).

But fortunately one can overcome both of these obstacles, as I will sketch in the following. This will allow us to transfer essential parts of the astounding computing and learning capability of Boltzmann machines to networks of spiking neurons with noise. In particular, it opens the door to using the parallel computing capability of networks of spiking neurons in continuous time in a substantially more efficient manner. Whether biologically more realistic neuron models such as spiking neurons could also provide, in technological applications, advantages over the simpler units of Boltzmann machines (beyond their power efficiency) is still a largely open question. Some first positive results that emerge in [31] will be discussed in Section II-E. In addition, the dendritic tree of biologically more truthful neuron models facilitates a neural network emulation of Bayesian networks (see Section II-D), as shown in [13].

The transition from Boltzmann machines to biologically more realistic networks of spiking neurons with noise also allows us to see the structure of cortical microcircuits in a new light. As an example, I will sketch how one prominent motif of cortical microcircuits, winner-take-all (WTA) circuits [see Fig. 3(b)], that consists of interacting excitatory and inhibitory neurons, provides important advantages for computing and learning in a stochastic context. These benefits of clever biological circuit architectures could not be addressed properly in Boltzmann machines, because their restriction to symmetric weights makes them unsuitable for understanding specific computational roles of excitatory and inhibitory neurons in stereotypical microcircuit configurations.

## II. NEW PARADIGMS

I want to review here two types of computational applications of networks of spiking neurons with noise: probabilistic inference from knowledge stored in complex probability distributions, and in Section II-E and F, the generation of heuristic solutions for hard computational problems (even for problems that are NP-hard [32]). In addition, I will address in Section II-G and H benefits of noise for self-organization and learning.

### A. Storing Knowledge in Probability Distributions of Network States

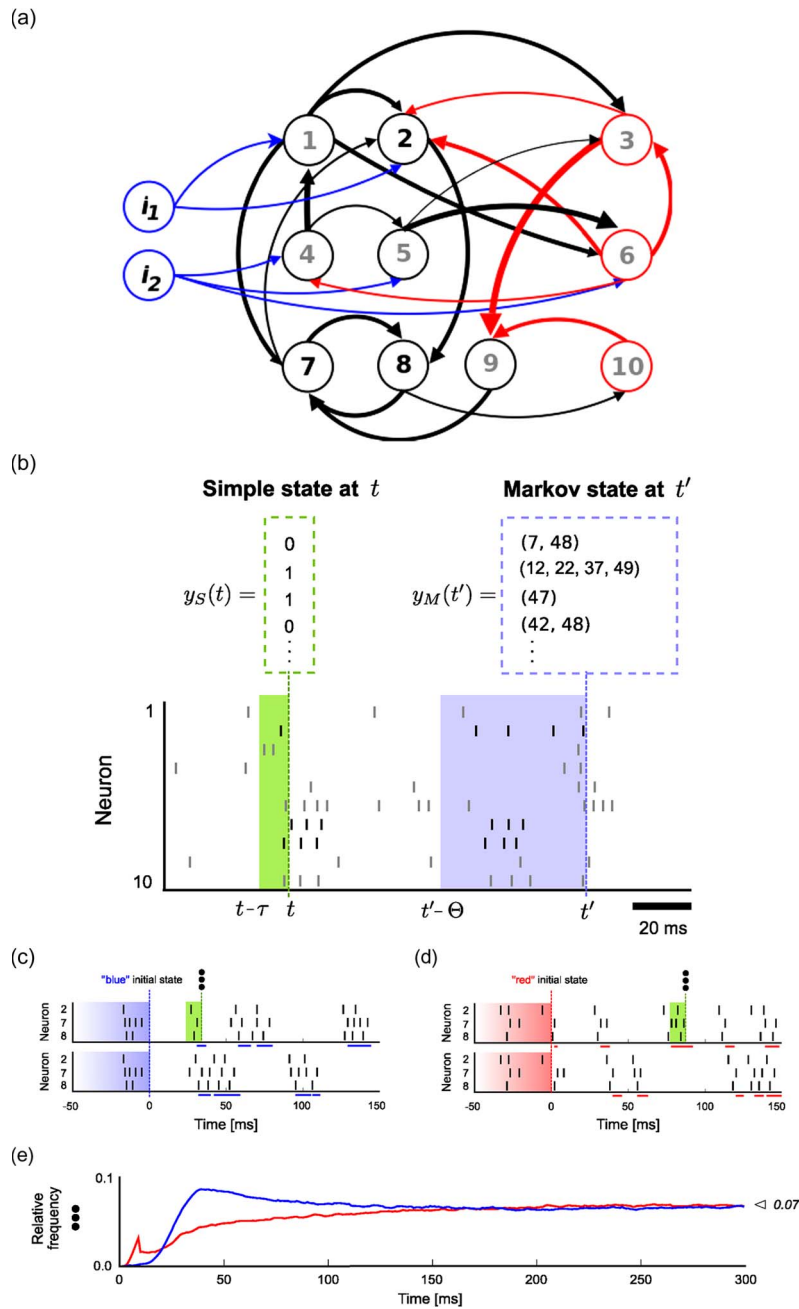
A fascinating new perspective of the way in which knowledge is encoded in the brain, and used for rapid decision making and motor control in response to complex multimodal sensory stimuli, was proposed in recent work in cognitive science (see, e.g., [33] and [34]). It has been hypothesized there that a large portion of long-term knowledge is encoded in the brain not in the form of facts and lists, like in a digital computer, but in the form of probability distributions  $p(z_1, \dots, z_m)$  over very large numbers  $m$  of behaviorally salient RVs  $z_i$ . These RVs can describe not only currently perceived sensory stimuli and

proprioceptive feedback, but also internal goals, learned knowledge, specific rules and strategies that are currently followed, as well as predictions and reward expectations for possible motor responses. This hypothesis is plausible from the perspective of artificial intelligence research, since it turned out to be very hard to encode real-world knowledge in the form of definite facts, and to use such deterministic knowledge for decision making through logical inference. A look at the current state of the art in artificial intelligence shows that these approaches have been largely abandoned, and replaced by more flexible probabilistic data structures and inference methods (see [35] and [36] for an account of this “probabilistic turn”). With regard to neural network research, this development has certainly increased the interest in possibilities for encoding probabilistic knowledge in neural networks.

If one accepts the hypothesis that knowledge is encoded in the brain in the form of probability distributions  $p(z_1, \dots, z_m)$ , the next question is how such distribution  $p$  could be encoded (and learned) by networks of spiking neurons. The more frequently proposed hypothesis is that such distributions are encoded there in an arithmetical form, and that knowledge is extracted from such probabilistic knowledge through arithmetical operations such as belief propagation [37], [38]. These operations are unfortunately very computation intense. Furthermore, any noise in neural networks would be detrimental for their implementation, since it reduces the reliability of deterministic data structures and inference methods.

An alternative hypothesis is that probability distributions  $p(z_1, \dots, z_m)$  are stored in the brain in an “embodied” form, through stochastic networks of spiking neurons that are able to generate samples from this distribution  $p$ . Probabilistic inference takes then the form of an analysis of a few samples (or exemplars) that are generated by such physical realization of  $p$ . Therefore, this approach does not require explicit calculations of probabilities. This is a frequently pursued approach for complex real-world probabilistic inference tasks in machine learning and artificial intelligence, where it is called MCMC sampling [27]. The complexity bottleneck of probabilistic inference is in this alternative approach shifted to the question what type of distributions  $p$  can be embodied by networks of spiking neurons with noise, and how fast they can generate samples from  $p$  that are not biased by the initial state of the network.

It turns out that basically any network  $C$  of spiking neurons that has a sufficient amount of noise (stochasticity), “embodies” a joint distribution  $p$  over a large number of RVs. For this purpose, network  $C$  is viewed as a Markov chain, i.e., as a stochastic system that moves according to stochastic laws from its current internal state to some other internal state. One can consider here several different notions of internal state of network  $C$ . An example for a simple notion of internal state is the green state shown in Fig. 1(b) for the simple circuit  $C$  of Fig. 1(a).



**Fig. 1. Illustration of network states and convergence to the stationary distribution of network states in a small network of spiking neurons.** (a) A small recurrent network of spiking neurons with noise consisting of ten neurons 1 . . . . 10 and two additional input neurons  $i_1, i_2$ . Neurons are colored by type (blue: input, black: excitatory, red: inhibitory). Line thickness represents synaptic weights. This circuit is a toy-size instantiation of the data-based cortical microcircuit model shown in Fig. 2(a), with neurons on three layers. (b) Notions of network state considered in this paper. Markov states (blue) are defined by the exact timing of all recent spikes within some longer time window  $\Theta$ , shown here for  $\Theta = 50$  ms. Simple states (green) only record which neurons fired recently (0 = no spike, 1 = at least one spike within a short window  $\tau$ , with  $\tau = 10$  ms throughout this figure). (c) Illustration of trial-to-trial variability in the occurrence frequency of a specific (partial) network state (1,1,1) for three selected neurons 2, 7, 8 of the circuit in (a). Two trials starting from identical initial network states  $y_M(0)$  are shown. Blue bars at the bottom of each trial mark periods where the subnetwork of neurons 2, 7, 8 was in simple state (1,1,1) at this time  $t$ . Note that the "blue" initial Markov state is shown only partially: it is actually longer [as in (b), but with  $\Theta = 1$  s] and comprises all neurons in the network. (d) Two trials starting from a different ("red") initial network state. Red bars denote occurrences of state (1,1,1) for "red" trials. (e) Convergence to the stationary distribution  $p_c$  in this small cortical microcircuit model is fast and independent of the initial state. This is illustrated for the relative frequency of simple state (1,1,1) within the first 300 ms after input onset. The blue/red line shows at each time  $t$  the relative frequency of simple state (1,1,1) for neurons 2, 7, 8, estimated from many ( $10^5$ ) "blue"/"red" trials. The relative frequency of simple state (1,1,1) rapidly converges to its stationary value regardless of the initial state (blue/red). This figure is a variation of a figure that has previously been published in [16] under the Creative Commons Attribution License (CCAL).

It has as many dimensions as there are neurons in  $C$ , and it records at any moment  $t$  which of the neurons of  $C$  has fired within the preceding time window of length  $\tau$  (it has value 1 in each dimension that corresponds to a neuron that has fired within this time window; otherwise 0). Such simple binary vector states are often considered in the analysis of experimental data in neuroscience; see, e.g., [39] for short lengths of time window  $\tau$  (in order to limit the amount of information loss arising when a neuron fires several times within such time window). One can rigorously prove that network (alias Markov chain)  $C$  has a unique stationary distribution  $p_C$  of such network states, to which it converges exponentially fast from any initial state. This result holds for a wide variety of neuron and synapse models, even data-based models with complex dendritic processing and other nonlinearities [16]. In particular, it holds for the simple model of a stochastic neuron defined through (4). It also holds for various models for noise, including models where noise is primarily generated through unreliable synaptic release. From the perspective of hardware models for networks of spiking neurons with noise, it is of interest that this result is likely to hold also for most hardware implementations of networks of spiking neurons with noise, in spite of numerous individual mismatches and implementation-induced nonlinearities.

This mathematical result is somewhat analogous to standard results on the existence of stationary distributions for Markov chains with finite sets of states and discrete time. But it requires a mathematically more complex framework in order to deal with an infinite (even continuous) set of states and with continuous time. Several variations of the notion of a network state are of interest in this context, such as the blue state shown in Fig. 1(b). There length  $\Theta$  of the time window is deliberately chosen to be larger than  $\tau$ , in order to ensure that the future spiking activity of the network after time  $t$  becomes (approximately) statistically independent of its past before time  $t - \Theta$ , given the exact protocol of all spike times during the time window from time  $t - \Theta$  to time  $t$ . Then, the blue state shown in Fig. 1(b) becomes a Markov state of the underlying Markov chain with continuous time.  $\Theta$  has to be chosen for that purpose larger than all internal time constants of membranes and synapses. It would have to be chosen to be infinitely large to make this rigorously true, but practically a value in the range of one or a few seconds suffices for most currently considered models for biological neurons and synapses. The result on the existence and uniqueness of a stationary distribution  $p_C$ , as well as exponentially fast convergence to it, is first derived for this more complex notion of network state. But once this result is proven, it implies the existence of a stationary distribution (and exponentially fast convergence to it) for any simpler notion of network state that arises by deleting information from the more complex Markovian notion of network state. In particular, it holds also for the simple network states indicated by the green state in Fig. 1(b).

This result also holds for a notion of network state with any even larger window length  $\Theta$ . The general result implies then that network  $C$  has a stationary distribution of really long sequences of firing activity, in particular for sequences or trajectories of simpler (see green state) network states. This additional result is of interest, because recent experimental data suggest that networks of neurons in the brain often go through certain stereotypical trajectories (or sequences) of simple network states [40], [41]; see also the discussion in [42].

A key question for extracting knowledge from any stationary distribution of network states is the following: Assume that the network starts in some more or less arbitrary network state at time  $t_0$  (and possibly in addition some stationary external input is activated at time  $t_0$ ). How long does one have to wait until the network produces samples from its *stationary* distribution of network states? This waiting time (called “convergence time” or “burn-in time” in MCMC sampling) is a key factor for determining the computation time that is needed for carrying out probabilistic inference through sampling (see Section II-B). This problem is illustrated in Fig. 1(c)–(e). We consider here for simplicity just the state defined by three neurons of the network, whose spikes are marked in black in Fig. 1(b). All time points  $\tau$  where all of these three neurons have fired since time  $t - \tau$  are marked in blue Fig. 1(c), and in red in Fig. 1(d). Two trial runs with the same initial state (a “blue” Markov state) are shown in Fig. 1(c). The blue curve in Fig. 1(e) indicates the relative frequency with which each time point  $t$  had this property for 10 000 repetitions of this experiment. One sees that it converges after about 150 ms to a fixed value 0.07, which is then the value of this (partial) network state under the stationary distribution  $p_C$  of simple network states of  $C$ . This experiment is repeated in Fig. 1(d) with a different initial state (the “red” state), and the corresponding statistics is shown by the red curve in Fig. 1(e). Both curves converge to the same value after about 150 ms, independently of whether the network started in the blue or red initial state. This fact is implied by the previously described theoretical result.

This is a very cumbersome way of evaluating empirically the speed of convergence of the distribution of network states produced by the network (alias Markov chain)  $C$  to its stationary distribution. In fact, it does not even achieve that if one considers just two initial states. But a number of more efficient heuristic methods have been developed in machine learning and statistics for estimating the speed of convergence to the stationary distribution. One frequently used method is the Gelman–Rubin test, which evaluates at any time  $t$  the quotient  $\hat{R}$  of the variance of the estimated frequency of network states between two trials with different initial states and the variance within two time windows for a single trial. If this quotient approaches values below 1.2, one usually says in machine learning that a Markov chain has converged (one

should note, however, that this is a heuristic measure; see the more detailed discussion in [16]).

Fig. 2 shows results of applications of this Gelman–Rubin test for evaluating the convergence speed for larger networks of stochastic neurons that are generated from the data-based template in Fig. 2(a). This panel shows the empirically measured interconnection profile of excitatory (black) and inhibitory (red) neurons on three laminae of the neocortex (layers 2/3 combined, layer 4, layer 5), with typical external input streams (“bottom up” input stream 1 is carrying information about sensory inputs; input stream 2 comes from other cortical areas). This microcircuit template is based on results of about 10 000 paired recordings from neurons of these six populations in the Lab of Alex Thomson in the Department of Pharmacology at the University of London and the Lab of Henry Markram at the Brain Mind Institute of EPFL (see [43] for more detailed credits, and for computational properties of this model). These experiments provided the connection probability between neurons [given in percent in Fig. 2(a)], average strength of existing synaptic connections (in terms of PSP amplitudes in millivolts; the thickness of arrows is scaled according to the product of these amplitudes with the connection probability), and the short-term dynamics of synaptic connections. For the simulations considered in Fig. 2, each neuron was simulated as a stochastically firing neuron according to (4).

Typical resulting network activity for an instantiation of the template from Fig. 2(a) with 560 neurons is shown in Fig. 2(b). Fig. 2(c) shows that the estimation of marginal probabilities for the stationary distribution  $p_C$  converges for such network  $C$  within about 100 ms. Furthermore, this panel shows the surprising result that this convergence speed does not depend on the size of network  $C$ , provided it is generated from the underlying data-based template shown in Fig. 2(a) (i.e., connections probabilities and distributions of synaptic parameters according to the model from [43]). Furthermore, Fig. 2(d) shows a corresponding result for a marginal distribution of 30 (instead of 1) neurons in the network. Altogether these simulation results suggest that the convergence speed for this data-based microcircuit model is relatively fast (in the range of one or several hundred milliseconds), and hence within the time span allotted to most computations carried out in cortical networks of neurons. This empirical result is, as one might have expected, not subsumed by the mathematically proven exponentially fast convergence to the stationary distribution for any such network  $C$ , because there are large constant factors in these exponential bounds. This empirical result is so far limited to these networks  $C$  (and a few other networks considered in [16]). Obviously, the structure and amount of noise, as well as the network architecture influence the convergence speed. According to theoretical results in [36], the convergence speed is related to the “conductance” of a Markov chain, i. e., the minimum probability with which it moves from one part of some (rather) arbitrary partition of its state space to

the other. This probability can, for example, be made arbitrarily small (and, hence, convergence speed arbitrary slow) in a stochastic neural network with several attractors, provided that the probability of escaping from an attractor is made sufficiently small. In other words, the more deterministic a neural network behaves, the slower is its convergence speed. But the exact nature of these dependencies for networks of spiking neurons remains to be determined.

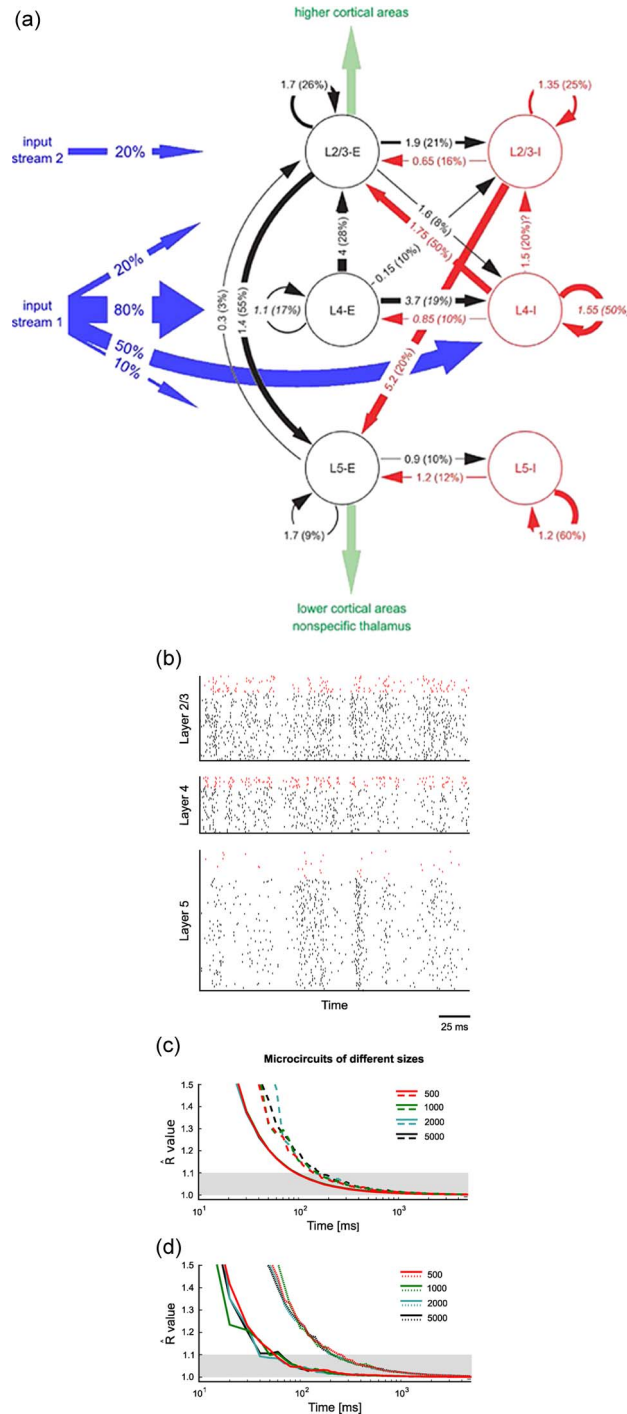
## B. Extracting Knowledge From an Embodied Probability Distribution Through Probabilistic Inference

The previously sketched results on the existence of a stationary distribution  $p_C$  of network states in large classes  $C$  of networks of spiking neurons with noise may help us to understand biological phenomena such as the highly structured spontaneous activity of brain networks [44], [45]. In addition, the simulation results of Figs. 1 and 2 provide examples for the extraction of knowledge from  $p_C$  through probabilistic inference via sampling.

These experiments were carried out under the assumption that network  $C$  receives during the sampling process external inputs  $\mathbf{x}$ . These may arise in the brain from sensory stimulation [input stream 1 in Fig. 2(a)] or top–down information [input stream 2 in Fig. 2(a)]. Results on stationary distributions of Markov chains are usually only considered for the case when these external inputs  $\mathbf{x}$  are stationary, e.g., fixed firing rates  $e$  of neurons that inject external inputs. In the language of probabilistic inference, such (momentary) stationary input is usually referred to as *evidence*, and the computation of  $p_C(E|\mathbf{x} = e)$  for some event  $E$  that can be defined in terms of network states  $\mathbf{y}$  is a typical case of probabilistic inference. For the simplest case when event  $E$  just concerns the firing of a specific single neuron  $\nu$ , this amounts to the estimation of a marginal probability  $p_C(\text{neuron } \nu \text{ fires}|\mathbf{x} = e)$ .

Surprisingly, even this very simple form of probabilistic inference (where one computes the marginal probability for a single RV) is, in general, very demanding from the computational perspective. In fact, estimating this probability within any error margin  $< 1/2$  is already NP-hard (see [36, Th. 9.4]). This implies that no deterministic algorithm, such as for example any variation of belief propagation (message passing), can provide such estimate within a practically interesting number of computation steps for distributions over a larger number of RVs. This computational difficulty becomes easier to understand if one takes into account that

$$\begin{aligned}
 p_C(\text{neuron } \nu \text{ fires} | \mathbf{x} = \mathbf{e}) &= \sum_{\text{network states } \mathbf{y}: \nu \text{ fires in } \mathbf{y}} p(\mathbf{y}|\mathbf{x} = \mathbf{e}) \quad (5)
 \end{aligned}$$



**Fig. 2.** Fast convergence of marginals of single neurons and more complex marginal distributions in cortical microcircuit models. (a) Data-based cortical microcircuit template for the interconnectivity of excitatory and inhibitory neurons on three cortical layers from [43]. (b) Typical spike response of an instantiation of the microcircuit model from (a) consisting of 560 stochastic point neurons. Spikes of inhibitory neurons are indicated in red. (c) Gelman–Rubin convergence diagnostic (see Section II-A) was applied to the marginals of single neurons (simple states,  $\tau = 10$  ms) in instantiations of the microcircuit template from (a) of different sizes (500–5000 neurons). Mean (solid lines) and worst marginal convergences (dashed lines) of single neurons are hardly affected by network size. (d) Convergence properties of frequencies of joint states of subsets of neurons in networks of different sizes. Dotted line: multivariate Gelman–Rubin analysis was applied to a subpopulation of 30 neurons [five neurons were chosen randomly from each of the six pools of neurons shown in (a)]. Solid line: convergence of a “random readout” neuron which receives spike inputs from 500 randomly chosen neurons in the microcircuit model. A remarkable finding is that in all these cases the network size does not affect convergence speed. Panels (b)–(d) are from a figure that has previously been published in [16] under the CCAL. Panel (a) is reprinted from [43] with kind permission of the Oxford University Press.

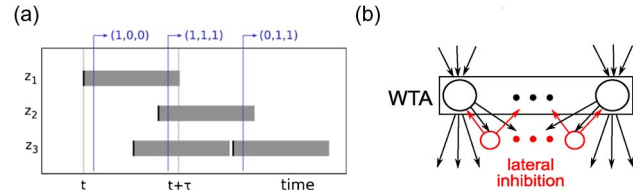


and that the number of network states  $\mathbf{y}$  over which this sum runs is exponential in the size of network  $C$ . On the other hand, the estimation of this probability is surprisingly simple in a sampling-based approach: one just observes how often neuron  $\nu$  fires within some time window. In other words, marginals can be estimated by simply ignoring the activity of the other neurons in network  $C$ . Of course, there is no free lunch in computational complexity, and NP-hardness implies that there exists the risk that we make (for any fixed length of the observation time window) a large error in our estimate of the firing rate of neuron  $\nu$ . But factors in favor of sampling are, on the one hand, that sampling of network states uses in a very efficient way the parallel computing capabilities of large networks of neurons. Furthermore, this sampling process takes place in continuous time, thereby avoiding a slowdown that is necessarily entailed by any discretization of time through a global clock. One sees the important role of continuous time for efficient probabilistic inference through sampling also clearly in software simulations, where *event-driven* simulators (such as [46]) are needed in order to avoid a slowdown. Hence, neuromorphic hardware that operates in continuous time may have here a particular advantage.

However, even these practically favorable factors do not provide guarantees, and occasional errors have to be expected. In other words, the empirical analysis of convergence speeds as carried out in Figs. 1 and 2 is an indispensable tool for estimating the computational efficiency of probabilistic inference through sampling. Furthermore, the concrete structure of noise (e.g., its amplitude and possible correlations between different local noise sources) and of the network (e.g., its locality of connections) have direct influences on this convergence speed. This computational efficiency perspective provides a completely new (and important) way of analyzing noise in spike-based networks that had previously not been explored.

### C. Which Probability Distributions Can Be Embodied by Networks of Spiking Neurons?

In order to understand the computational capability of networks of spiking neurons with noise, we need to understand which probability distributions  $p$  could potentially be embodied as stationary distribution  $p_C$  of a network  $C$  of spiking neurons. For that purpose, we need to adopt some convention for relating spiking activity to values of RVs. One straightforward approach is to assume that each spike of a neuron sets a corresponding binary RV for some short time (e.g., for the typical temporal extensions of an EPSP, say 20 ms) from value 0 to value 1; see Fig. 3(a). Note that this convention is closely related to the role of a neuron in defining the simple state of a network [see green state in Fig. 1(b)]. An easy way of encoding the value of a multinomial RV through spiking neurons is provided by WTA circuits where at (almost) any moment in time just one of the excitatory neurons fires



**Fig. 3.** Standard conventions for relating spiking activity to discrete values of RVs of a joint distribution  $p$ . (a) Binary RVs. Assume that each binary RV  $z_k$  of  $p$  is represented by a spiking neuron  $\nu_k$ , and a spike of  $\nu_k$  (marked in black) corresponds to setting  $z_k = 1$  for a short time period (marked in gray). (b) Multinomial RVs. Assume that each neuron in a WTA circuit votes for a specific value of the variable, and that this value changes as soon as some other neuron in the WTA circuit fires.

[see Fig. 3(b)]. Thus, one can relate each of these excitatory neurons to one possible value of an RV.

For the sake of simplicity, I will consider in the following just binary RVs. The question that needs to be examined then is which joint distributions  $p(z_1, \dots, z_m)$  over binary RVs  $z_k$  can be embodied through the stochastic dynamics of networks of spiking neurons. Positive results are provided by the neural sampling theory of [12]. It is shown there that a sufficient condition for reproducing a given distribution  $p$  (over binary RVs) as the stationary distribution  $p_C$  of a network  $C$  of spiking neurons with noise is given by the neural computability condition (NCC) presented as follows.

For each RV  $z_k$ , there is some neuron  $\nu_k$  whose firing probability density at time  $t$ , if it is not in a refractory period, is

$$\rho_k(t) = \frac{1}{\tau} \times \frac{p(z_k = 1 | z_{\setminus k}(t))}{p(z_k = 0 | z_{\setminus k}(t))} \quad (6)$$

where  $z_{\setminus k}$  denotes the other RVs besides  $z_k$ .

The proof of this result is instructive, since it shows how some results on stationary distributions for reversible Markov chains (such as Boltzmann machines) can be ported to certain nonreversible Markov chains (such as recurrent networks of spiking neurons). One gets mathematically rigorous results only for an idealized type of spiking neuron, but numerical simulations suggest that the error is not large for biologically more realistic models (see [12] for details).

Furthermore, it is shown in [12] that this sufficient condition readily implies that any distribution  $p$  over binary RVs with at most second-order dependencies [these are the so-called Boltzmann distributions, which can always be represented in the form of (3)] can be embodied by networks of spiking neurons with noise. It is well known that also distributions with higher order dependencies can be encoded by Boltzmann distributions with the help of

auxiliary RVs. However, it was shown in [13] that this way of emulating higher order dependencies tends to slow down probabilistic inference through sampling in a quite drastic manner. The reason for that is that the auxiliary RVs have to be defined by Boolean formulas in terms of the original binary RVs, and neural sampling in the presence of such logically strict (rather than probabilistic soft) dependencies between RVs tends to slow down the speed of probabilistic inference through sampling (it tends to reduce the conductance of the underlying Markov chain by creating partitions of the state space with low transition probability from one set of states to the other, since several RVs have to change *simultaneously* for certain state changes).

#### D. Probabilistic Inference in Arbitrary Bayesian Networks Through Networks of Spiking Neurons With Noise

An alternative method for embodying distributions over binary RVs with arbitrary dependencies is laid out in [13]. One can assure the validity of the NCC by assigning to each binary RV  $z_k$  not only a spiking neuron  $\nu_k$  that represents its value, but also an auxiliary network  $A_k$  of spiking neurons that directly ensures the NCC for  $z_k$ . For that purpose, the auxiliary network  $A_k$  tests which pattern of values is currently encoded by the firing activity of the neurons assigned to the other RVs  $z_i$  with  $i$  different from  $k$ , and then creates an input to neuron  $\nu_k$  that produces a firing probability for it that agrees with the value given by the NCC. It is shown in [13] for various examples that this method produces embodiments of distributions  $p$  with higher order dependencies as unique stationary distributions of networks of spiking neurons with noise, for which the distribution of network states converges from any initial state relatively fast to this stationary distribution.

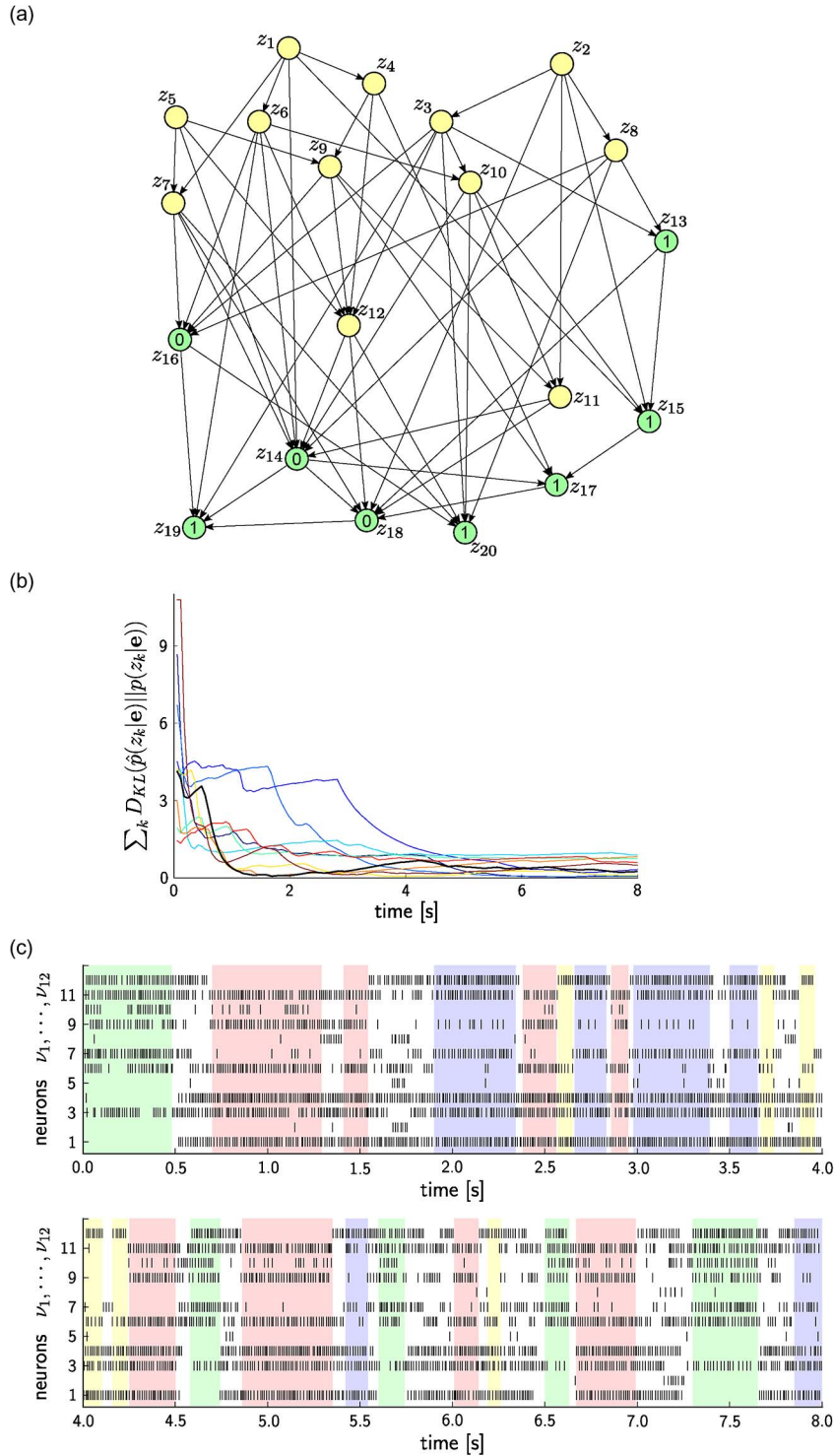
One common method for constructing and describing probability distributions with higher order dependencies are Bayesian networks [see Fig. 4(a) for an example]. These are directed acyclic graphs where each node is labeled by an RV. The edges of the graph capture from an intuitive perspective causal dependencies between RVs. More precisely, the edges capture the dependency structure (conditional independence) among the RVs: each RV (node) is independent from the RVs on all nondescendant nodes, given the values of the RVs at its parent nodes. Another way of characterizing the relationship between Bayesian networks and probability distributions is the following: A Bayesian network (as a directed graph) completely specifies a joint probability distribution over the RVs that label its nodes once one specifies for each node (RV) its conditional probability table, given all possible assignments of values to the RVs that label its parent nodes. The resulting joint probability distribution is then the product over all these conditional distributions.

But one has to be careful: Bayesian networks are, in general, very different from neural networks since they do not define a dynamics or network computation. They “just” represent certain factorization (or, equivalently, conditional independence) properties of probability distributions. Only in rare special cases (such as the one discussed in this section) one can directly associate a (stochastic) neural network  $N$  with a Bayesian network  $B$  so that  $N$  represents, through its stationary distribution of network states, the same probability distribution as  $B$  (once  $B$  is augmented by conditional probability tables for all its nodes, as described above).

Whenever more than a single edge ends at a node of a Bayesian network, a term of order larger than 2 arises in the probability distributions that it represents. This situation is often described in terms of the “explaining away effect,” which occurs, for example, in context dependence perception tasks (see [13]), and can, therefore, not be ignored for real-world applications. Many Bayesian networks that arise in real-world applications even contain undirected cycles. Standard belief-propagation algorithms have no guarantee to converge to the correct solution for such Bayesian networks (see the discussion in [13] and [36]). Hence, the noise-driven approach of [13] based on sampling is of particular relevance for carrying out probabilistic inference for such tasks. Higher order dependencies among RVs, as well as undirected cycles of Bayesian networks can be handled in this approach via suitable preprocessing circuits  $A_k$  in the spiking network (as described above).

An instance of a relatively large Bayesian network with numerous nodes where several edges and several undirected cycles converge is shown in Fig. 4(a). The green nodes represent RVs for which evidence is entered for a generic probabilistic inference task. The goal is to estimate the resulting (posterior) marginal probabilities for the other RVs. The fact that evidence is entered primarily at nodes toward the bottom of the Bayesian network is typical for real-world applications, where one wants to identify likely causes for observations. The estimation of marginal probabilities for the other RVs (at the yellow nodes) requires then to explore various possible causes for the observations. First simulation experiments in [13] suggest that, even for this fairly large and complex Bayesian network, probabilistic inference through sampling in a corresponding network of spiking neurons tends to be quite fast, in the range of a few seconds [see Fig. 4(b)]. Fig. 4(c) illustrates the typical behavior of the neurons  $\nu_k$  in the network that encode possible causes (i.e., the RVs at yellow nodes): they move forth and back between various characteristic network states (each indicated by a different color) that correspond to joint assignments to these RVs that have relatively high probability under the given joint distribution  $p$ .

Curiously enough, such stochastic switching between different network states is also observed in experimental



**Fig. 4. Emulation of probabilistic inference through sampling in spiking networks for a fairly large and complex Bayesian network with numerous converging edges and undirected cycles. (a) The Bayesian network. Evidence  $e$  is entered for the RVs at the green nodes. (b) The sum of the differences (measured by the Kullback-Leibler divergences) between the correct and estimated marginal posterior probabilities for each of the unobserved RVs, calculated from the generated samples (spikes) from the beginning of the simulation up to the current time indicated on the x-axis. Separate curves with different colors are shown for each of the ten trials with different initial conditions (randomly chosen). The bold black curve corresponds to the simulation for which the spiking activity is shown in (c). (c) The spiking activity of the 12 neurons that represent the binary RVs on the yellow nodes in (a) during the simulation from  $t = 0$  s to  $t = 8$  s, for one of the ten simulations. The neural network enters and remains in different network states (indicated by different colors), corresponding to different modes of the posterior joint probability distribution. This figure was previously published in [13] under the CCAL.**

data from neuroscience, in particular, in fMRI recordings (hence, on a larger spatial and temporal scale) of spontaneous activity in the so-called default network of the brain [44]. This default network has been implicated as being essential both for memory retrieval and for imagination [47], thereby creating a link to Section II-E. Bayesian networks and other graphical models have also been proposed as functional models for motor control, where some RVs denote joint positions, and others (for which typically evidence is entered for solving motor control tasks through probabilistic inference) denote movement goals, expected rewards, and contingencies (e.g., obstacles). This topic will be addressed in Section II-F.

### E. Solving Constraint Satisfaction Problems With Networks of Spiking Neurons

Many computationally difficult problems are constraint satisfaction problems (CSPs). CSPs are familiar to all of us, since real-world tasks such as planning a route to the office, planning a family vacation, or managing our budget requires to satisfy as many constraints as possible from a large number of more or less important constraints. Some of these constraints are of an uncertain nature (will there be a traffic jam on the shortest route?), but have to be rationally dealt with nevertheless. Computational complexity theory has taught us that many practically important CSPs belong to the class of NP-hard problems [32], which means that there is practically no hope of solving these problems on a Turing machine (or any classical digital computer) within a number of computation steps that does not increase exponentially with the size of the problem. This means that even for moderately sized problems (in terms of the bit length of encoding the problem) of say 100, the number of computation steps that a digital computer would need is astronomical. NP-hardness has thwarted our attempts to endow digital computers with general capabilities for logical inference or related types of artificial intelligence. In particular, there is no practically useful computer program available for deciding whether some concrete Boolean formula is implied by some other Boolean formulas. This lack of algorithmic solutions follows from the NP-hardness of the satisfiability problem, i.e., of deciding whether a given Boolean formula can be satisfied.

I will argue in the following that networks of spiking neurons with noise provide new opportunities for solving fairly large instances of NP-hard and other CSPs, at least heuristically (in principle, one could also implement deterministic approximation algorithms for solving hard CSPs in deterministic networks of spiking neurons, but I am not aware of publications on that). First results in [16] and [31] suggest that these networks can often produce, within a reasonable time span, an approximate solution that satisfies a large portion of the given constraints. This approach will not provide miracle solutions to NP-hard

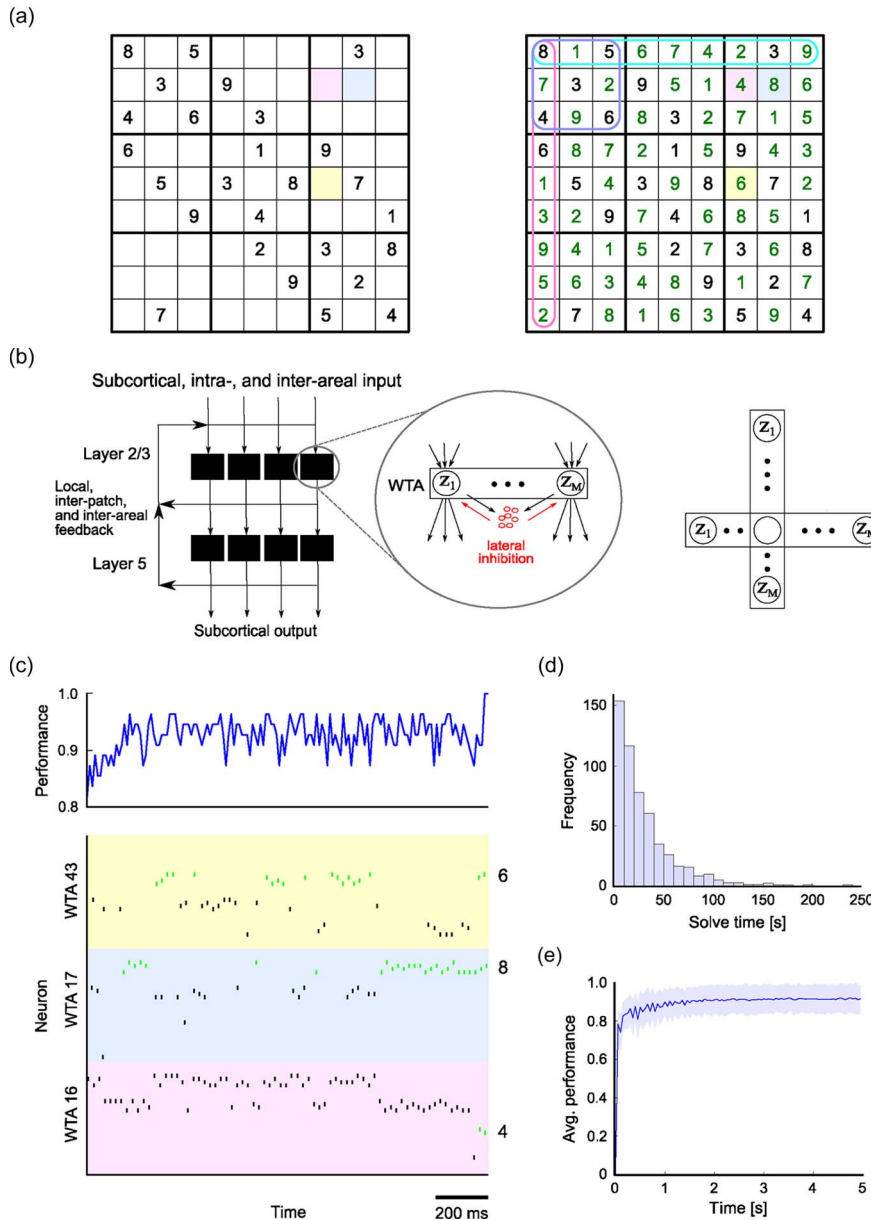
problems, but it will engage specific advantages of networks of spiking neurons with noise, such as their immense parallelity and their asynchronous operation in continuous time, in a particularly efficient way. Thereby, they may advance the frontier for practically relevant computational problems that can be solved within a given time and energy budget.

Whenever a neuron fires, it changes, for a moment, the firing probability of other neurons to which it is synaptically connected. Therefore, each synaptic connection and each network motif can be viewed as a constraint on the network states  $\mathbf{y}$  that appear with high probability under the stochastic dynamics of a network of spiking neurons with noise. In other words, the autonomously generated stochastic dynamics of a network of spiking neurons with noise solves an immensely complex constraint satisfaction problem. It generates network states  $\mathbf{y}$  that are most of the time consistent with a large portion of the interlocking and circular local constraints that are imposed by the architecture of a recurrent network of neurons and its parameters. Hence, if we manage to program constraints into the architecture of a network of spiking neurons that are meaningful for the solution of a practically relevant constraint satisfaction problem, and if we find ways of controlling the frequency of network states  $\mathbf{y}$  during the resulting stochastic dynamics of the network in dependence of the number and importance of satisfied constraints that the network state  $\mathbf{y}$  represents, we have found a new way of using networks of spiking neurons for purposeful computations.

Research in this direction is still very young. It builds on and expands earlier related work on Boltzmann machines [25], which did not offer the promise of energy-efficient computation that networks of spiking neurons hold, but which had already pioneered the use of networks of stochastic computing elements for solving CSPs.

I will first demonstrate the capability of networks of spiking neurons to solve CSPs with the well-known Sudoku puzzle. Sudoku serves well as an example because it demands substantial problem solving capability. The rules of the Sudoku game can be easily encoded into a recurrent network of WTA circuits. Each WTA circuit consists of an ensemble of excitatory neurons with lateral inhibition [see Figs. 3(b) and 5(b)]. It induces higher order dependencies between RVs, and hence cannot be readily implemented by a Boltzmann machine.

Each excitatory neuron can, in fact, be a part of several interlocking WTA motifs [see the right part of Fig. 5(b)]. This architecture makes it easy to impose the interlocking constraints of Sudoku (and of many other constraint satisfaction problems). Each excitatory neuron votes for placing a particular digit into an empty field of the grid. But this vote is subject to the constraints that only one digit can be placed into this field, and that each digit  $1, \dots, 9$  occurs only once in each column, in each row, and in each  $3 \times 3$  subgrid marked in Fig. 5(a). Hence, each excitatory



**Fig. 5. Solving Sudoku, a constraint satisfaction problem, through structured interactions between stochastically firing excitatory and inhibitory neurons. (a)** A “hard” Sudoku puzzle with 26 given numbers (left). The solution (right) is defined uniquely by the set of givens and the additional constraints that each digit must appear only once in each row, column, and  $3 \times 3$  subgrid. **(b)** An implementation of the constraints of the Sudoku game in a spiking neural network  $C$  consists of a recurrent network of WTA circuits, resembling somewhat a previously proposed simple model for cortical circuits [48]. The same excitatory neuron  $v_k$  (representing a binary RV  $z_k$ ) can be a part of several such WTA circuits (right). Each digit in a Sudoku field is associated with an excitatory neuron which votes for this digit when it emits a spike (actually, each RV  $z_k$  was represented in this simulation by four excitatory neurons, but this is an implementation detail). Each excitatory neuron participates in four WTA circuits, corresponding to the constraints that only one digit can be active in each Sudoku field, and that a digit can appear only once in each row, column, and  $3 \times 3$  subgrid. **(c)** A typical network run for the Sudoku puzzle from (a) is shown during the last 1500 ms before the correct solution was found [the total solve time was approximately 3 s in this run; see (d) for statistics of solve times]. The network performance (fraction of active neurons that vote for correct values) over time is shown at the top. The spiking activity is shown for three (out of 81) WTA circuits associated with the three colored Sudoku fields in (a). In each of these WTA circuits, there are 36 excitatory neurons (nine digits and four pyramidal cells for each digit). Spikes are colored green for those neurons which code for the correct digit in “their” Sudoku field (6, 8, and 4 in the example). **(d)** Histogram of solve times (the first time the correct solution was found) for the Sudoku from (a). Statistics were obtained from 1000 independent runs. The sample mean is 29 s. **(e)** Average network performance for this Sudoku converges quickly during the first five seconds to a high value of 0.9, corresponding to 90% correctly found digits (average taken over 1000 runs; shaded area:  $\pm 2$  standard deviations). Thereafter, among all  $9^{81}$  possible configurations, the network spends most time in good approximate solutions. The correct solution occurs particularly often, on average approximately 2% of the time (not shown). This figure is a variation of a figure that has previously been published in [16] under the CCAL.

neuron is simultaneously a part of four WTA circuits. A specific puzzle can be entered by providing strong input  $\mathbf{x}$  to those neurons which represent the given numbers in Sudoku [see the left part of Fig. 5(a)]. This initiates a quite intuitive dynamics: “Clamped” neurons start firing strongly, and as a consequence, neurons which code for conflicting digits in the same Sudoku field, the same row, column, or  $3 \times 3$  subgrid become strongly inhibited. In those cases where inhibition leaves more than one possible digit open, a tentative digit will be picked randomly by those neurons which happen to fire first among their competitors in the stochastic network dynamics. The stochasticity of this process ensures that, instead of getting stuck in local optima, the network automatically explores several potential configurations whenever multiple possibilities remain. Altogether, through this combination of constraint enforcement and random exploration, those network states which violate few constraints (i.e., good approximate solutions) are visited with much higher probability than states with many unsatisfied constraints. A theoretical framework for this analysis and methods for shaping this probability distribution are developed in [31]. In our simulations, we found that the solve time (the time until the correct solution is found for the first time) generally depends on the hardness of Sudoku, in particular, on the number of givens. For the “hard” Sudoku with 26 givens from Fig. 5(a), solve times are approximately exponentially distributed with an average of 29 s [Fig. 5(d)]. The run of the resulting network of spiking neurons shown in Fig. 5(c) exhibits on the side another nice property of this computational paradigm for networks of spiking neurons: The network provides, at any time, *some* heuristic solution of the problem, rather than requiring the user to wait for a certain “computation time” before it provides any useful output. If, however, there is more time for an answer, the system keeps producing better solutions. This capability, which is obviously quite desirable for real-time systems, is sometimes referred to as *anytime computing*.

It would be interesting to compare the performance of different neural network approaches—in particular, deterministic versus stochastic networks—on a common set of Sudoku puzzles. But this work has not yet been carried out. An interesting deterministic approach for solving Sudoku in nonspiking deterministic neural networks via linear programming had been introduced in [49]. The approach in [16] is apparently the only one so far with spiking neurons (deterministic or stochastic). Its performance can be improved with the help of a new method from [31]: internal temperature control (see below, not published).

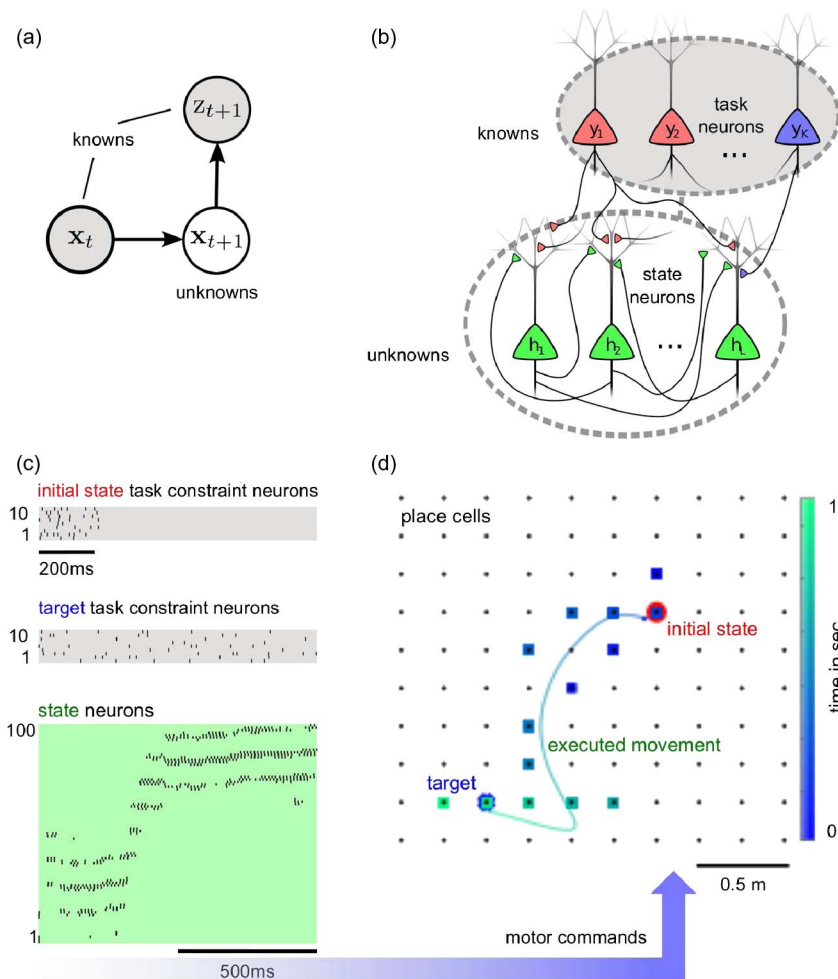
It has been shown that Sudoku is, in principle, NP-hard (provided that one considers puzzles with increasing numbers of fields). In a forthcoming work [31], it is shown that networks of spiking neurons can also solve other well-known NP-complete problems, such as the

satisfiability problem and the traveling salesman problem. Furthermore, it turns out that there exists, for certain architectures, a theoretical framework that allows us to estimate the probability of network states  $\mathbf{y}$  in terms of the number of constraints which they satisfy, similarly as for Boltzmann machines. But in contrast to Boltzmann machines, we can work here also with asymmetric synaptic weights (as needed already to interconnect an excitatory neuron with an inhibitory neuron). It turns out that one can exploit, in addition, another asymmetry of spike-based networks that is not encountered in Boltzmann machines: The asymmetry between the generation of a spike and its postsynaptic effects in other neurons (which results from a stochastic process) on the one hand, and the end of this signal (which is largely deterministic, given when it started) on the other hand. It is shown in [31] that this asymmetry is useful for exploration in stochastic search. In addition, new methods are exhibited in [31] for modulating “temperature”  $T$  of the distribution of network states through mechanisms within the network, e.g., for increasing the difference between the probabilities  $p_C(\mathbf{y}|\mathbf{x})$  of network states  $\mathbf{y}$  in dependence of the number of constraints which they satisfy. A distribution with smaller differences enhances exploration. Furthermore, one can also design networks in such a way that they lock (with very high probability) into an optimal solution, once they have found it. These results can be viewed as first steps toward a new design methodology for solving hard computational tasks through networks of spiking neurons with noise.

## F. Applications of Networks of Spiking Neurons With Noise to Movement Planning

Many motion control and planning problems in robotics can be formulated as probabilistic inference tasks [50]. In this setup, the previously acquired experience (“procedural knowledge”) is formalized as a probability distribution over RVs that may, for example, represent the current state, the next state, possible actions, goals, rewards, and contingencies. The given “evidence” can, in this case of probabilistic inference, not only represent the current state, but also currently active contingencies (e.g., obstacles), desired rewards, or the intended next state. The most likely action for achieving this is then inferred through probabilistic inference. This perspective is also currently pursued as a possible way of understanding biological motor control and planning [51].

The methods that I have sketched in the preceding sections allow us to implement this probabilistic inference approach in networks of spiking neurons with noise. The pros and cons of this approach are explored in current research [52], [53]. A simple demonstration is provided in Fig. 6. It shows that networks of spiking neurons are not only able to “infer” the next action, but also to play through whole movement plans. The underlying stochastic sampling approach induces a significant trial-to-trial variability of movement plans, which is qualitatively similar



**Fig. 6. Movement planning through probabilistic inference in a Bayesian network, implemented through a network of spiking neurons with noise. (a) Part of the Bayesian network with RVs  $x_t$  and  $x_{t+1}$  representing the state (position) at steps  $t$  and  $t + 1$ , and RVs  $z_{t+1}$  (“task variables”) representing constraints and goals for this step. Nodes that represent RVs whose values are given (such as  $z_{t+1}$ ) or have already been inferred (such as  $x_t$ ) are shaded in gray. All these RVs are represented by populations of spiking neurons shown in (b). (b) Part of the network of spiking neurons that implements the Bayesian network from (a). (c) Firing activity of all neurons during a single trial. (d) Resulting planned movement trajectory. Figure drawn by Elmar Rueckert, and printed with his permission.**

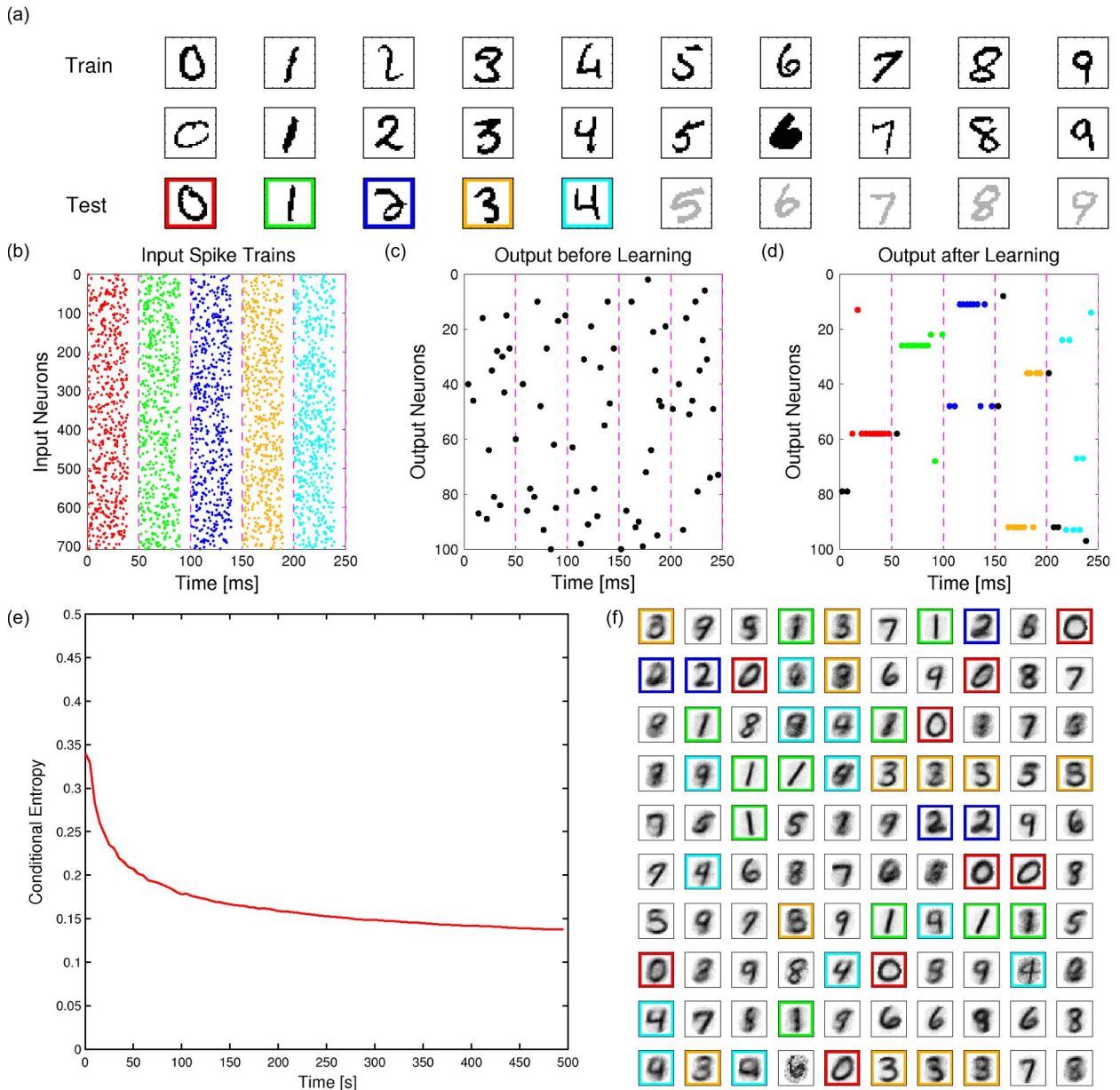
to experimental data from biological organisms. This variability also provides an exploration mechanism that is needed for reinforcement learning (see Section II-H).

### G. Role of Noise for Self-Organization Processes in Networks of Spiking Neurons

It is well known that self-organization processes without noise are prone to get stuck in local minima. Hence, one can expect that noise will be beneficial for a variety of self-organization mechanisms of networks of spiking neurons. I want to focus here on the discussion of one example, where one can see that noise can endow networks of spiking neurons with the capability to create and maintain through synaptic plasticity subnetworks that carry out specific computational tasks, in this case through probabilistic inference. Also, here, there exist attractive

paradigms from Boltzmann machines, where it had been demonstrated that suitable synaptic plasticity rules endow them with the capability to create (without any supervision or rewards) internal models for the probability distribution  $p^*$  of input patterns which they receive. In other words, Boltzmann machines can learn to become generative models that can not only learn to categorize input patterns without any guidance by a supervisor, but also generate typical input patterns with a distribution similar to  $p^*$  (see, e.g., [29]). This generative aspect is an essential component of many powerful learning methods. Such inversion of the direction of processing is easy for Boltzmann machines since their synaptic connections are required to be symmetric, hence they are “backwards drivable.”

Unfortunately, such an inversion of the processing direction that one needs for a generative learning approach



**Fig. 7. Emergent discrimination of handwritten digits through STDP in a WTA circuit with noise.** (a) Examples of digits from the MNIST data set. The third row contain test examples that had not been shown during learning via STDP. (b) Spike train encoding of the first five samples in the third row of (a). Each pixel is encoded for 30 ms by a spike train with a high or low firing rate (corresponding to a black or white pixel). Colors illustrate the different classes of digits. (c) and (d) Spike trains produced by the  $K = 100$  competing neurons of the WTA circuit before and after learning with STDP for 500 s. Colored spikes indicate that the class of the input and the class for which the neuron is mostly selective [based on human classification of its generative model shown in (f)] agree, otherwise spikes are black. (e) Speed of the self-organization process of the 100 output neurons. (f) Internal models generated by STDP for the 100 neurons of the WTA circuit after 500 s. The network had not received any information about the number of different digits that exist (the colors for different ways of writing the first five digits were assigned by the human supervisor). On the basis of this assignment, the test samples in row 3 of (a) had been recognized correctly. This figure was previously published in [15] under the CCAL.

is rather foreign to networks of spiking neurons. However, it has recently been shown that noise endows networks of spiking neurons even without this inverse processing capability (and without the requirement of symmetric weights) with similarly powerful unsupervised learning

methods as some traditional generative models [14], [15]. The most commonly studied and experimentally supported rule for (unsupervised) synaptic plasticity in networks of spiking neurons is spike-timing-dependent plasticity (STDP); see, e.g., [54]. According to this rule, a synaptic



weight is increased whenever the presynaptic neuron fires shortly before the postsynaptic neuron, and decreased whenever the presynaptic neuron fires shortly after the postsynaptic neuron (hence, STDP selects and strengthens those synaptic connections that appear to be causally involved in the firing of the postsynaptic neuron). It has been shown in [14] and [15] that STDP can in the presence of noise install in WTA circuits the arguably most powerful method for unsupervised learning and self-organization that is currently known: expectation maximization (EM). More precisely, it is proven there that STDP implements, in these network motifs, a stochastic online approximation of EM for fitting a mixture of multinomial distributions (or of exponential family distributions [14]) to the input distribution  $p^*$ . In particular, it enables them to learn to carry out Bayesian inference: A prior is learned through (unsupervised) adaptation of the excitabilities (or firing thresholds) of the neurons, and the natural processing of spike inputs (in the usual bottom-up direction) can be viewed as sampling from the posterior of the resulting internally learned mixture distribution. An application of this stochastic learning method, which is called spike-based EM (SEM), to a distribution  $p^*$  of unlabeled handwritten digits from the Mixed National Institute of Standards and Technology (MNIST) database is demonstrated in Fig. 7.

The autonomously learned approximation to  $p^*$  in the form of a mixture of multinomials is analyzed for this example in Fig. 7(f), in terms of the implicit internal models (weights) of the competing neurons of the WTA circuit, projected back into the pixel input space. One sees that the network of spiking neurons has learned without any supervision not only “templates” for each handwritten digit, but also characteristic differential ways of writing each digit (since it does not know that a vertical stroke and a slanted stroke represent the same digit “1”). This self-organization of a network of 100 spiking neurons with lateral inhibition provides a nice demonstration of the self-organization capabilities of networks of spiking neurons with noise.

If one adds lateral *excitatory* connections between the neurons of the WTA circuit, and applies to them the same STDP rule as for bottom-up connections, the network learns to approximate another important probabilistic model: a hidden Markov model (HMM); see [55]. There, the bottom-up spike inputs (that represent observations in the terminology of HMMs) are combined by the autonomously learning neurons with spike inputs from their neighbors that provide a temporal context for characterizing the bottom-up spike input. Hence, the circuit implements forward sampling (in MCMC terminology).

If one moves to more complex networks of spiking neurons with noise that form a randomly connected network of several WTA circuits (which happens to be a pretty good rough approximation to the structure of a generic cortical column; see [48]) one sees that, in these

networks, similar dynamic memory traces for repeatedly encountered external inputs emerge as those that have been found in experimental data from various parts of sensory cortices and parietal cortex: characteristic assemblies and sequences of assemblies [42]. It is commonly conjectured that these are the traces of long-term memory in the cortex [41], [56].

In current work, these autonomously learning spiking networks are brought one step closer to biological models for cortical columns by taking into account that pyramidal cells on superficial and deep layers of cortex form an “infinite” sheet of excitatory neurons with local lateral inhibition, rather than an array of isolated WTA circuits [57]. Furthermore, the lateral inhibition is examined more closely in [58], where symbolic lateral inhibition is replaced by interactions with concrete types of inhibitory neurons. Moving to a more relaxed form of such lateral inhibition creates sparse WTA (sWTA) circuits, where more than a single neuron can fire simultaneously [59]. This relaxed lateral inhibition endows the competing neurons with the capability to focus each on some *partial* feature of a salient spike input pattern, rather than forcing each neuron to become an “expert” for a whole input pattern.

These results open the door to a rich world of architectures where noise endows networks of spiking neurons with powerful self-organization capabilities. In contrast to many other demonstrations of self-organization capabilities of neural networks, one can analyze these models with analytic methods, at least approximately, similarly as for Boltzmann machines. But the mathematical formalism which is required for that turns out to differ in several aspects from the simpler one used for Boltzmann machines.

In the recent work by Pecevski and Maass [60], such network learning capability is considered in the context of architectures for networks of spiking neurons that emulate probabilistic inference in arbitrarily given Bayesian networks (see Section II-D). One sees there that each auxiliary network  $A_k$  for a neuron  $\nu_k$  that is supposed to classify the current firing pattern of other neurons  $\nu_i$  can learn, in the presence of noise through STDP, to capture the most frequently occurring ones of such patterns. Therefore, it is no longer necessary to set the parameters of the auxiliary networks  $A_k$  by hand, and their size is no longer crucial. Hence, the goal to create network architectures for spiking neurons that can autonomously learn to build an internal Bayesian network model for complex external input streams is one step closer.

## H. Role of Noise for Reinforcement Learning in Networks of Spiking Neurons

Reinforcement learning is learning by trial and error, hence it requires active exploration. This holds also for learning through reward-based synaptic plasticity (see, e.g., [61] and [62]), and it is only possible if the neurons or

synapses are noisy. In fact, the noise level is critical, since if there is too much noise, the credit assignment (where those synaptic connections that were active shortly before a reward are strengthened) becomes less precise. If there is too little noise, the network may not be able to find through exploration any network response that is rewarded.

A critical feature of any reward-based learning scenario is the number of trials that are required, and in this regard, models tend to stay behind the performance of many biological organisms. One aspect that may speed up reward-based learning is a clever exploration strategy, which means in this setting a suitable structure of stochastic variability in a network of neurons [63], [64]. We have seen in previous sections that networks of spiking neurons are, in principle, able to represent a large variety of distributions of network states. It is an interesting open question how this distribution should be structured so that the stochastic dynamics of the network supports fast exploration of possible solutions for concrete learning tasks.

### III. DISCUSSION

This paper has reviewed a new perspective for encoding knowledge in networks of spiking neurons: Network  $C$  of spiking neurons with noise is able to encode substantial amounts of knowledge in its stationary distribution  $p_C$  of network states that it embodies through its stochastic dynamics. In fact, depending on the precise definition of network state, different salient distributions can be represented simultaneously by network  $C$ , including distributions of longer firing sequences of neurons (as reviewed in [56] and [41]) that could encode, for example, procedural knowledge. In addition, network  $C$  has the capability to encode in the presence of a background oscillation *different* distributions of simple states for *different* phases of this oscillation [16]. Such phase coding is apparently also used by the brain [65], [66]. It is conjectured [67] that neural synchrony plays in the brain an important role in information binding. Note that neural synchrony is, to a large extent, just a different way of analyzing the role of network states [especially of simple states as in Fig. 1(b)], rather than firing rates of single neurons. Hence, the concept and results sketched in Section II-A, B, and E provide a perspective where neural synchrony and information binding are integrated into a more general framework for stochastic network computations.

This method of encoding knowledge in networks of spiking neurons bypasses disputes about neural coding of single variables encoded by artificial stimuli (such as oriented gratings) with spike timing or rates (see, e.g., [38] and [68]–[71]) by moving to the more global perspective of network states that typically encode several behaviorally salient variables. This more global perspective coincides with the perspective of any downstream network that

receives synaptic input from tens of thousands of neurons. Furthermore, it allows to encode really powerful and demanding data structures in networks of spiking neurons with noise, such as graphical models for complex probability distributions (e.g., Bayesian networks), as I have sketched

These data structures have become, in current machine learning, computer vision, and artificial intelligence, a favored way of encoding real-world knowledge about likely causes of sensory inputs, and also more abstract relationships. In addition, graphical models are becoming useful in robotics for encoding knowledge about how to move in order to achieve specific goals in different contexts and under varying constraints (e.g., obstacles). In fact, one may argue that this probabilistic form of knowledge representation has replaced, in large parts of computer science and its applications, deterministic forms of knowledge representation. Deterministic knowledge representations, in combination with logical inference methods, turned out to be of limited use for generating intelligent responses and strategies for real-world tasks. For that, one needs to generate at high speed clever heuristic guesses and action choices on the basis of large numbers of uncertain observations and experiences, and very few reliable facts.

I have also reviewed results from the last few years, where it has been shown that knowledge can be extracted in a principled manner from data structures that are represented through distributions of network states: Through probabilistic inference (e.g., estimation of posterior marginal probabilities) via sampling from such distributions. A potential obstacle for this approach is the sampling time that it requires. The first investigation of this problem for generic models of cortical microcircuits in [16] suggests that short sampling times in the range of a few hundreds of milliseconds (of simulated biological time) are sufficient. Applied to hardware implementations that are substantially faster (e.g., with a speedup factor of 10 000 in the approach that is pursued in the Lab of Karlheinz Meier at the University of Heidelberg [72]) this would provide response times in the range of tens of microseconds. But this topic certainly requires further research. In particular, we need to understand which network architectures and what types of noise enhance the sampling speed of networks of spiking neurons.

Several recent reviews present arguments which support the hypothesis that the brain does, in fact, carry out probabilistic inference [34], [38], [51], [71], [73], [74]. Interesting counter-arguments have been collected in [75]. In particular, Bowers and Davis [75] cite results of experimental studies in neuroscience, such as [76] and [77] where it is shown that orientation-selective neurons in area V1 respond highly reliably for an oriented bar in their preferred direction. This is *per se* not an argument against the encoding of probabilities of external variables through

firing probabilities, since if the probability of an external variable is close to 1, the corresponding neuron should fire highly reliably under such a probabilistic coding scheme.

If probabilistic inference is carried out in the brain, it is not yet clear to what extent this is implemented through deterministic or stochastic neural circuit computations. Several interesting theoretical studies (see, e.g., [37], [78], and [79]) have shown that some elements of probabilistic inference that can be implemented through deterministic message passing (belief propagation) can, in principle, also be implemented by deterministic networks of spiking neurons (in fact, any deterministic computation, i.e., any Turing machine, can be simulated by a deterministic network of spiking neurons [80]).

One important advantage of approaches that employ noise as a resource becomes apparent if one analyzes the efficiency (in terms of the number of neurons that are used, and the required overhead for the organization of computations) of the resulting architectures and algorithms. In fact, some of the previously cited studies of possible neural implementations of probabilistic inference through belief propagation only consider relatively easy instances of probabilistic inference, such as applications of the Bayes rule, or inference in Bayesian networks without undirected cycles, or even without converging edges (i.e., without “explaining away” mechanisms).

Another strong argument for sampling-based approaches arises from the more learning-friendly aspect of this implementation strategy: Examples from an external multivariable distribution that needs to be learned by a network are of the same type as the samples that a stochastic network produces after learning [60]. This setup is consistent with an implementation of network learning through local learning rules such as Hebbian learning and STDP (possibly within a generative framework, and gated by global variables that signal saliency or reward expectations). This is hard to achieve if conditional probabilities have to be encoded explicitly (as analog numbers) through neural codes, which is typically required by implementations of probabilistic inference through message passing (belief propagation). The review in [38] mentions an interesting argument against the encoding of probabilities through firing rates in V1: the width of orientation tuning curves is independent of contrast in anesthetized cats [81]. It is proposed in [81] that this effect results from stochastic resonance. Hence, it may very well be the case that noise has a different functional role in neural circuits that specialize in feature extraction in primary sensory cortices [10].

Furthermore, the neural sampling perspective provides a model for higher level brain processes such as resting activity [44], [45], [73], stochastic switching between brain states for ambiguous stimuli (see the discussion in [82] and [12]), imagination and movement planning [83] that are difficult to explain or model with a deterministic model for

brain computation. One should also note that the multiplexing and context dependence of neural codes that has recently been exhibited for higher brain areas [84], [85] is predicted by models for implementing probabilistic inference in networks of spiking neurons, such as those that were sketched in Section II-A–II-C. The underlying experimental protocols induce dependencies among random variables that represent stimulus features, selected behavior, and context information. Hence such dependencies have to be expected from neurons that encode these random variables through their firing probability under a stationary distribution  $p_C$  of network states.

The predictions of nondeterministic models for computation and learning in the brain can soon be tested more rigorously with new experimental methods (such as high-density silicon probes, calcium imaging) that capture responses of large populations of neurons in several cortical areas during realistic active behavior, also over weeks and months (and not only for artificially impoverished stimuli, or after overtraining). In fact, an analysis of the statistics of resulting high-dimensional network states (and trajectories of network states) appears to become one of the most promising methods for analyzing and understanding the resulting experimental data (see [39] for a first step in this direction, and [86] and [87] for difficulties in the data analysis).

Independently from the exact nature of brain computations, it is clear that the new concepts and methods that have been reviewed in this paper will provide a boost in computational power and learning capability to artificial networks of spiking neurons that manage to treat noise as a resource. Furthermore, current work on solving NP-complete constraint satisfaction problems through networks of spiking neurons with noise [59] suggests in addition that an internal control of the noise profile will be beneficial. Thus, noise engineering for networks of spiking neurons seems to become an exciting new research area.

The brain has more processors (i.e., neurons) than a typical supercomputer, but consumes less than 30 W of energy. A major goal of computer technology is to port some of this astounding energy efficiency into nanoscale circuits for future generations of computer hardware. Since noise is inevitable at this scale, a key breakthrough would be achieved by exploiting noise as a resource for computation and learning in future computing hardware. First steps in this direction are described in [4]. ■

## Acknowledgment

The author would like to thank S. Habenschuss, Z. Jonke, R. Legenstein, D. Pecevski, D. Potzinger, and E. Rueckert for scientific advice and help with the figures. He would also like to thank three anonymous reviewers for helpful suggestions.

## REFERENCES

- [1] M. McDonnell, K. Boahen, A. Ijspeert, and T. Sejnowski, "Engineering intelligent electronic systems based on computational neuroscience," *Proc. IEEE*, 2014, to be published.
- [2] B. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J. M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proc. IEEE*, 2014, to be published.
- [3] T. Pfeil, A. Grübl, S. Jeltsch, E. Müller, P. Müller, M. A. Petrovici, M. Schmuker, D. Brüderle, J. Schemmel, and K. Meier, "Six networks on a universal neuromorphic computing substrate," *Front. Neurosci.*, vol. 7, no. 11, 2012, DOI: 10.3389/fnins.2013.00011.
- [4] T. J. Hamilton, S. Afshar, A. van Schaik, and J. Tapon, "Stochastic electronics: A neuro-inspired design paradigm for integrated circuits," *Proc. IEEE*, 2014, DOI: 10.1109/JPROC.2014.2310713, to be published.
- [5] J. G. Borst, "The low synaptic release probability *in vivo*," *Trends Neurosci.*, vol. 33, no. 6, pp. 259–266, 2010.
- [6] H. Markram and M. Tsodyks, "Redistribution of synaptic efficacy between neocortical pyramidal neurons," *Nature*, vol. 382, no. 6594, pp. 807–810, 1996.
- [7] A. A. Faisal, L. P. J. Selen, and D. M. Wolpert, "Noise in the nervous system," *Nature Rev. Neurosci.*, vol. 9, no. 4, pp. 292–303, 2008.
- [8] Y. Yarom and J. Hounsgaard, "Voltage fluctuations in neurons: Signal or noise?" *Physiol. Rev.*, vol. 91, no. 3, pp. 917–929, 2011.
- [9] P. G. Clarke, "The limits of brain determinacy," *Proc. Biol. Sci.*, vol. 279, no. 1734, pp. 1665–1674, 2012.
- [10] M. D. McDonnell and L. M. Ward, "The benefits of noise in neural systems: Bridging theory and experiment," *Nature Rev. Neurosci.*, vol. 12, no. 7, pp. 415–426, 2011.
- [11] A. Destexhe and M. Rudolph-Lilith, *Neuronal Noise*. New York, NY, USA: Springer-Verlag, 2012.
- [12] L. Buesing, J. Bill, B. Nessler, and W. Maass, "Neural dynamics as sampling: A model for stochastic computation in recurrent networks of spiking neurons," *PLoS Comput. Biol.*, vol. 7, no. 11, 2011, e1002211.
- [13] D. Pecevski, L. Büsing, and W. Maass, "Probabilistic inference in general graphical models through sampling in stochastic networks of spiking neurons," *PLoS Comput. Biol.*, vol. 7, no. 12, 2011, e1002294.
- [14] S. Habenschuss, H. Pühr, and W. Maass, "Emergence of optimal decoding of population codes through STDP," *Neural Comput.*, vol. 25, no. 6, pp. 1371–1407, 2013.
- [15] B. Nessler, M. Pfeiffer, L. Buesing, and W. Maass, "Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity," *PLOS Comput. Biol.*, vol. 9, no. 4, 2013, e1003037.
- [16] S. Habenschuss, Z. Jonke, and W. Maass, "Stochastic computations in cortical microcircuit models," *PLoS Comput. Biol.*, vol. 9, no. 11, 2013, e1003311.
- [17] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Oxford, U.K.: Oxford Univ. Press, 2005.
- [18] A. Alaghi and J. P. Hayes, "Survey of stochastic computing," *ACM Trans. Embedded Comput. Syst.*, vol. 12, no. 2s, 2013, DOI: 10.1145/2465787.2465794.
- [19] A. Litwin-Kumar and B. Doiron, "Slow dynamics and high variability in balanced cortical networks with clustered connections," *Nature Neurosci.*, vol. 15, no. 11, pp. 1498–1505, 2012.
- [20] M. M. Churchland and L. F. Abbott, "Two layers of neural variability," *Nature Neurosci.*, vol. 15, no. 11, pp. 1472–1474, 2012.
- [21] T. Tetzlaff, M. Helias, G. T. Einevoll, and W. Diesmann, "Decorrelation of neural-network activity by inhibitory feedback," *PLoS Comput. Biol.*, vol. 8, no. 8, 2012, e1002596.
- [22] M. A. Petrovici, J. Bill, I. Bytschok, J. Schemmel, and K. Meier, "Stochastic inference with deterministic spiking neurons," 2013. [Online]. Available: arXiv: 1311.3211
- [23] G. E. Hinton and T. J. Sejnowski, "Optimal perceptual inference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, B. D. Ripley, Ed., 1983, pp. 448–453.
- [24] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cogn. Sci.*, vol. 9, no. 1, pp. 147–169, 1985.
- [25] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*. New York, NY, USA: Wiley, 1989.
- [26] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, 1982.
- [27] C. Andrieu, N. D. Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Mach. Learn.*, vol. 50, no. 1–2, pp. 5–43, 2003.
- [28] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Jun. 1984.
- [29] R. Salakhutdinov and G. Hinton, "An efficient learning procedure for deep Boltzmann machines," *Neural Comput.*, vol. 24, no. 8, pp. 1967–2006, 2012.
- [30] R. Jolivet, A. Rauch, H. Lüscher, and W. Gerstner, "Predicting spike timing of neocortical pyramidal neurons by simple threshold models," *J. Comput. Neurosci.*, vol. 21, no. 1, pp. 35–49, 2006.
- [31] Z. Jonke, S. Habenschuss, and W. Maass, "Noise enables networks of spiking neurons to carry out efficient stochastic search in high-dimensional spaces," 2014, to be published.
- [32] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: Freeman, 1979.
- [33] E. Vul and H. Pashler, "Measuring the crowd within probabilistic representations within individuals," *Psychol. Sci.*, vol. 19, no. 7, pp. 645–647, 2008.
- [34] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman, "How to grow a mind: Statistics, structure, abstraction," *Science*, vol. 331, no. 6022, pp. 1279–1285, 2011.
- [35] M. Oaksford and N. Chater, *Bayesian Rationality: The Probabilistic Approach to Human Reasoning*. Oxford, U.K.: Oxford Univ. Press, 2007.
- [36] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques (Adaptive Computation and Machine Learning)*. Cambridge, MA, USA: MIT Press, 2009.
- [37] A. Steimer, W. Maass, and R. Douglas, "Belief-propagation in networks of spiking neurons," *Neural Comput.*, vol. 21, pp. 2502–2523, 2009.
- [38] A. Pouget, J. M. Beck, W. J. Ma, and P. E. Latham, "Probabilistic brains: Knowns and unknowns," *Nature Neurosci.*, vol. 16, pp. 1170–1178, 2013.
- [39] P. Berkes, G. Orban, M. Lengyel, and J. Fiser, "Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment," *Science*, vol. 331, pp. 83–87, 2011.
- [40] A. Luczak, P. Barthó, S. L. Marguet, G. Buszáki, and K. D. Harris, "Sequential structure of neocortical spontaneous activity *in vivo*," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 1, pp. 347–352, 2007.
- [41] G. Buzsáki, "Neural syntax: Cell assemblies, synapses, readers," *Neuron*, vol. 68, no. 3, pp. 362–385, 2010.
- [42] S. Klampfl and W. Maass, "Emergence of dynamic memory traces in cortical microcircuit models through STDP," *J. Neurosci.*, vol. 33, no. 28, pp. 11515–11529, 2013.
- [43] S. Hauesler and W. Maass, "A statistical analysis of information processing properties of lamina-specific cortical microcircuit models," *Cereb. Cortex*, vol. 17, no. 1, pp. 149–162, 2007.
- [44] M. E. Raichle, "Two views of brain function," *Trends Cogn. Sci.*, vol. 14, no. 4, pp. 180–190, 2010.
- [45] M. H. Mohajerani, A. W. Chan, M. Mohsenvand, J. LeDue, R. Liu, D. A. McVea, J. D. Boyd, Y. T. Wang, M. Reimers, and T. H. Murphy, "Spontaneous cortical activity alternates between motifs defined by regional axonal projections," *Nature Neurosci.*, vol. 16, no. 10, pp. 1426–1435, 2013.
- [46] D. Pecevski, D. Kappel, and Z. Jonke, "NEVESIM: Event-based neural simulation framework with a python interface," *Front. Neuroinf.*, 2013, to be published.
- [47] D. L. Schacter, D. R. Addis, D. Hassabis, V. C. Martin, R. N. Spreng, and K. K. Szpunar, "The future of memory: Remembering, imagining, the brain," *Neuron*, vol. 76, pp. 677–694, 2012.
- [48] R. J. Douglas and K. A. Martin, "Neuronal circuits of the neocortex," *Annu. Rev. Neurosci.*, vol. 27, pp. 419–451, 2004.
- [49] J. J. Hopfield, "Searching for memories, sudoku, implicit check bits, the iterative use of not-always-correct rapid neural computation," *Neural Comput.*, vol. 20, pp. 1119–1164, 2008.
- [50] M. Toussaint and C. Goerick, "A Bayesian view on motor control and planning," in *From Motor to Interaction Learning in Robots. Studies in Computational Intelligence*, O. Sigaud and J. Peters, Eds. New York, NY, USA: Springer-Verlag, 2010, pp. 227–252.
- [51] M. Botvinick and M. Toussaint, "Planning as inference," *Trends Cogn. Sci.*, vol. 16, no. 10, pp. 485–488, 2012.
- [52] D. Kappel, D. Pecevski, S. Chase, A. Whitford, and W. Maass, "A model for closed-loop probabilistic motor control through networks of spiking neurons with noise," 2014, to be published.
- [53] E. Rueckert, D. Pecevski, D. Kappel, and W. Maass, "An implementation of movement planning as probabilistic inference through networks of spiking neurons with noise," 2014, to be published.
- [54] N. Caporale and Y. Dan, "Spike timing-dependent plasticity: A Hebbian

- learning rule," *Annu. Rev. Neurosci.*, vol. 31, pp. 25–46, 2008.
- [55] D. Kappel, B. Nessler, and W. Maass, "STDP installs in winner-take-all circuits an online approximation to hidden Markov model learning," *PLoS Comput. Biol.*, 2014.
- [56] A. Luczak and J. N. MacLean, "Default activity patterns at the neocortical microcircuit level," *Front. Integr. Neurosci.*, vol. 6, no. 30, 2012, DOI: 10.3389/fnint.2012.00030.
- [57] J. Bill, L. Buesing, S. Habenschuss, B. Nessler, R. Legenstein, and W. Maass, "Distributed Bayesian computation and self-organized learning in sheets of spiking neurons with local lateral inhibition," 2014, to be published.
- [58] S. Haessler and W. Maass, "Inhibitory networks orchestrate the self-organization of computational function in cortical microcircuit motifs through STDP," 2014, to be published.
- [59] Z. Jonke, R. Legenstein, S. Habenschuss, and W. Maass, "Emergent coding of features and Bayesian computation in ensembles of pyramidal cell with relaxed lateral inhibition," 2014, to be published.
- [60] D. Pecevski and W. Maass, "Learning in an uncertain world: STDP installs in networks of spiking neurons complex memory structures that facilitate probabilistic inference," 2014, to be published.
- [61] R. Legenstein, D. Pecevski, and W. Maass, "A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback," *PLoS Comput. Biol.*, vol. 4, no. 10, 2008, e1000180.
- [62] G. M. Hoerzer, R. Legenstein, and W. Maass, "Emergence of complex computational structures from chaotic neural networks through reward-modulated Hebbian learning," *Cereb. Cortex*, 2012, DOI: 10.1093/cercor/bhs348.
- [63] D. J. Herzfeld and R. Shadmehr, "Motor variability is not noise, but grist for the learning mill," *Nature Neurosci.*, vol. 17, no. 2, pp. 149–150, 2014.
- [64] H. G. Wu, Y. R. Miyamoto, L. N. G. Castro, B. P. Ölveczky, and M. A. Smith, "Temporal structure of motor variability is dynamically regulated and predicts motor learning ability," *Nature Neurosci.*, vol. 17, no. 2, pp. 312–321, 2014.
- [65] G. Dragoi and G. Buzsáki, "Temporal encoding of place sequences by hippocampal cell assemblies," *Neuron*, vol. 50, no. 1, pp. 145–157, 2006.
- [66] M. Siegel, M. R. Warden, and E. K. Miller, "Phase-dependent neuronal coding of objects in short-term memory," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 50, pp. 21341–21346, 2009.
- [67] W. Singer, "Cortical dynamics revisited," *Trends Cogn. Sci.*, vol. 17, no. 12, pp. 616–626, 2013.
- [68] T. D. Sanger, "Neural population codes," *Current Opinion Neurobiol.*, vol. 13, pp. 238–249, 2003.
- [69] Y. Huang and R. P. Rao, "Predictive coding," *Wiley Interdisciplinary Rev., Cogn. Sci.*, vol. 2, no. 5, pp. 580–593, 2011.
- [70] M. Boerlin, C. K. Machens, and S. Denève, "Predictive coding of dynamical variables in balanced spiking networks," *PLoS Comput. Biol.*, vol. 9, no. 11, 2013, e1003258.
- [71] T. Lochmann and S. Denève, "Neural processing as causal inference," *Current Opinion Neurobiol.*, vol. 21, no. 5, pp. 774–781, 2011.
- [72] J. Schemmel, D. Bruderle, A. Grubl, M. Hock, K. Meier, and S. Millner, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2010, pp. 1947–1950.
- [73] J. Fiser, P. Berkes, G. Orban, and M. Lengyel, "Probabilistic computation: A possible functional role for spontaneous activity in the cortex," *Perception*, vol. 40, pp. 53–53, 2011.
- [74] T. L. Griffiths, N. Chater, D. Norris, and A. Pouget, "How the Bayesians got their beliefs (and what those beliefs actually are): Comment on Bowers and Davis," *Psychol. Bull.*, vol. 138, pp. 415–422, 2012.
- [75] J. S. Bowers and C. J. Davis, "Bayesian just-so stories in psychology and neuroscience," *Psychol. Bull.*, vol. 138, no. 3, pp. 389–414, 2012.
- [76] M. Gur, A. Beylin, and D. M. Snodderly, "Response variability of neurons in primary visual cortex (v1) of alert monkeys," *J. Neurosci.*, vol. 17, pp. 2914–2920, 1997.
- [77] M. Gur and D. M. Snodderly, "Response variability of neurons in primary visual cortex (v1) of alert monkeys," *J. Neurosci.*, vol. 17, no. 8, pp. 2914–2920, 2006.
- [78] R. P. N. Rao, "Bayesian computation in recurrent neural circuits," *Neural Comput.*, vol. 16, pp. 1–38, 2004.
- [79] S. Denève, "Bayesian spiking neurons I: Inference," *Neural Comput.*, vol. 20, no. 1, pp. 91–117, 2008.
- [80] W. Maass, "Lower bounds for the computational power of networks of spiking neurons," *Neural Comput.*, vol. 8, no. 1, pp. 1–40, 1996.
- [81] J. S. Anderson, I. Lampl, D. C. Gillespie, and D. Ferster, "The contribution of noise to contrast invariance of orientation tuning in cat visual cortex," *Science*, vol. 290, no. 5498, pp. 1968–1972, 2000.
- [82] S. J. Gershman, E. Vul, and J. B. Tenenbaum, "Multistability and perceptual inference," *Neural Comput.*, vol. 24, no. 1, pp. 1–24, 2012.
- [83] B. E. Pfeiffer and D. J. Foster, "Hippocampal place-cell sequences depict future paths to remembered goals," *Nature*, vol. 497, no. 7447, pp. 74–79, 2013.
- [84] M. Rigotti, O. Barak, M. R. Warden, X. J. Wang, N. D. Daw, E. K. Miller, and S. Fusi, "The importance of mixed selectivity in complex cognitive tasks," *Nature*, vol. 497, no. 7451, pp. 585–590, 2013.
- [85] V. Mante, D. Sussillo, K. V. Shenoy, and W. T. Newsome, "Context-dependent computation by recurrent dynamics in prefrontal cortex," *Nature*, vol. 503, no. 7474, pp. 78–84, 2013.
- [86] M. Okun, P. Yger, S. L. Marguet, F. Gerard-Mercier, A. Benucci, S. Katzner, and K. D. Harris, "Population rate dynamics and multineuron firing patterns in sensory cortex," *J. Neurosci.*, vol. 32, no. 48, pp. 17 108–17 119, 2012.
- [87] J. Fiser, M. Lengyel, C. Savin, G. Orbn, and P. Berkes, "How (not) to assess the importance of correlations for the matching of spontaneous and evoked activity," 2013. [Online]. Available: arXiv:1301.6554

## ABOUT THE AUTHOR

**Wolfgang Maass** received the Ph.D. degree in mathematics from the Ludwig-Maximilians-Universität, Munich, Germany, in 1974.

He was a Postdoctoral Fellow at the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA; the University of Chicago, Chicago, IL, USA; and the University of California at Berkeley, Berkeley, CA, USA. He was on the faculty of the University of Illinois at Chicago, Chicago, IL, USA, and is now Professor of Computer Science at the Graz University of Technology, Graz, Austria, where he is also the Director of the Institute of Theoretical Computer Science. His earlier work had focused on the theory of computation, both from the perspective of mathematics and from the complexity-based perspective of computer science. He made there contributions in particular to the complexity theory of mathematical models for networks of neurons, and to the



design and analysis of learning algorithms. His primary current research interest are principles of brain computation, a topic for which he is also responsible in the ten-year European Union (EU) Flagship Project "Human Brain Project" that started in 2013. Relatively few computer scientists are currently working on this topic, and his Lab in Graz has become a focal point for their efforts in understanding brain computations and learning. He has authored over 200 scientific papers. In particular, he has coauthored publications on brain computation with a large number of experimental neuroscientists. Together with H. Markram and T. Natschlaeger, he had developed the liquid computing model that provides a computational paradigm for realistic models of cortical microcircuits that consist of many different types of neurons and synaptic connections. In recent years, his Lab has started to address also the ubiquitous trial-to-trial variability of experimentally observed neural responses, and has demonstrated that this feature could provide an important clue for the organization of brain computations.