SCIENTIFIC REPORTS

OPEN

Spiking neurons with short-term synaptic plasticity form superior generative networks

Luziwei Leng[1], Roman Martel[1], Oliver Breitwieser[1], Ilja Bytschok[1], Walter Senn[2], Johannes Schemmel[1], Karlheinz Meier[1] & Mihai A. Petrovici[1,2]

Spiking networks that perform probabilistic inference have been proposed both as models of cortical computation and as candidates for solving problems in machine learning. However, the evidence for spike-based computation being in any way superior to non-spiking alternatives remains scarce. We propose that short-term synaptic plasticity can provide spiking networks with distinct computational advantages compared to their classical counterparts. When learning from high-dimensional, diverse datasets, deep attractors in the energy landscape often cause mixing problems to the sampling process. Classical algorithms solve this problem by employing various tempering techniques, which are both computationally demanding and require global state updates. We demonstrate how similar results can be achieved in spiking networks endowed with local short-term synaptic plasticity. Additionally, we discuss how these networks can even outperform tempering-based approaches when the training data is imbalanced. We thereby uncover a powerful computational property of the biologically inspired, local, spike-triggered synaptic dynamics based simply on a limited pool of synaptic resources, which enables them to deal with complex sensory data.

Neural networks are, once again, in the focus of both the artificial and the biological intelligence communities. Originally inspired by the dynamics and architecture of cortical networks[1,2], they have increasingly strayed away from their biological archetypes, prompting questions about their relevance for understanding the brain[3,4]. However, their recent hardware-fueled dominance[5] has motivated renewed efforts to align them with biologically more plausible models[6–9]. Moreover, neural networks have been used to explain some aspects of *in-vivo* cortical dynamics[10,11].

Two questions are immanent to these efforts: From a machine learning perspective, how useful are spike-based versions of deep neural networks? And from a biological perspective, how much can we learn about the brain from artificial neural networks? Much of the recent work on neural networks has focused on the "forward" computation pathway, i.e., learning pattern classification through error backpropagation[12]. However, the "backward" pathway required for generative models has also made significant progress[13,14]. A key aspect to the success of a generative model is its capability to mix, i.e., travel through the probability landscape that it needs to represent. The performance gain of recently proposed models is to a large extent due to refined mixing algorithms, most of which are based on a form of simulated tempering[15–17].

The discriminative capacity of the neocortex is well-established, as evidenced by the difficulty of artificial systems to achieve superhuman classification performance[12]. Simultaneously however, the brain also appears to learn a generative model of its sensory environment[18–20]. How these capabilities are achieved remains an open question, but it is unlikely that complex tempering schedules are at work.

One mechanism that is capable of modulating synaptic weights and thereby shaping the probability landscape of a neural network is short-term plasticity (STP). In this work, we investigate the ability of this biologically ubiquitous mechanism to improve the mixing capabilities of generative neural networks. Furthermore, we show how hierarchical spiking networks endowed with STP can simultaneously become good discriminative and generative models, a feature that is difficult to achieve due to the conflicting nature of these two tasks. We thereby offer a potential explanation for the generative capabilities of cortical networks, while at the same time proposing a simple but efficient mechanism to bolster the usefulness of spiking networks for machine learning applications.

[1]Kirchhoff Institute for Physics, University of Heidelberg, Heidelberg, Germany. [2]Department of Physiology, University of Bern, Bern, Switzerland. Luziwei Leng and Mihai A. Petrovici contributed equally to this work. Correspondence and requests for materials should be addressed to M.A.P. (email: mpedro@kip.uni-heidelberg.de)
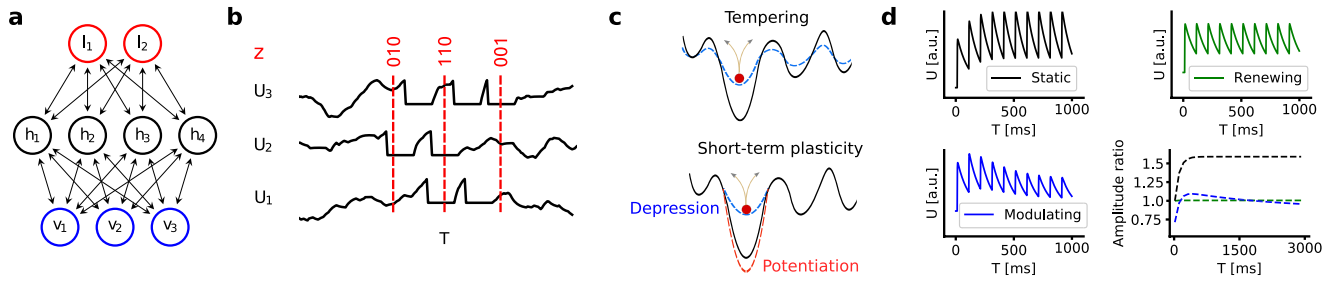
Figure 1. Network architecture and dynamics. (**a**) Structure of a hierarchical sampling spiking network. Its classical counterpart is a restricted Boltzmann machine with a visible (*v*), hidden (*h*) and label (*l*) layer. (**b**) Interpretation of states as samples in a spiking network. A neuron with a freely evolving membrane potential is said to be in the state $z_k = 0$ and switches to the state $z_k = 1$ upon firing, where it stays for the duration of the refractory period. (**c**) In order to correctly sample from a posterior distribution, a network needs to be able to mix, i.e., traverse barriers between low-energy basins. To facilitate mixing, tempering methods globally rescale the energy landscape with an inverse temperature (top). In contrast, STP can be viewed as only modulating the energy landscape locally, thereby only affecting the currently active attractor (bottom). (**d**) Sketch of the membrane potential evolution for three relevant scenarios: static (black), renewing (green) and modulated (blue) synapses. Bottom right: envelope of the PSP height for three parameter sets (U_0, τ_{rec}) from the manuscript: (1, 0) (black), (1, τ_{syn}) (green) and (0.01, 280 ms) (blue). Note how the latter only weakly modulates the PSP height.

This can be of particular interest in combination with spiking neuromorphic systems which, compared to conventional simulation platforms, implement fast and energy-efficient physical models of neuro-synaptic dynamics[21,22].

Network Architecture and Dynamics

We start with a brief introduction of Boltzmann machines as generative models and their spike-based implementation. We then describe the problem of mixing and outline the essential elements of tempering-based solutions. Finally, we discuss the model of STP that we later use in our spiking networks.

Boltzmann machines and spiking networks. Among the neural networks proposed as generative models for high-dimensional input, Boltzmann machines (BMs)[23] are arguably the most prominent[24–27]. Neurons in BMs are binary units with states $z_k \in \{0, 1\}$. These states are typically updated in a sequential schedule in a way that implements Gibbs sampling from a target Boltzmann distribution

$$p_B(z|W, b) = Z^{-1} \exp\left[-\beta E(z)\right] \qquad (1)$$

with the inverse temperature $\beta \in (0, 1]$, partition function Z and the energy function $E(z) = -z^T W z / 2 - z^T b$ parametrized by the weight matrix W and bias vector b. This is achieved by having each neuron compute a local "membrane potential" as the log-odds of its conditional firing probability, which for the Boltzmann distribution is equal to a weighted sum over input activities:

$$u_k = \ln \frac{p_B(z_k=1|z_{\setminus k})}{p_B(z_k=0|z_{\setminus k})} = \sum_{i \neq k} w_{ki}\, z_i + b_k. \qquad (2)$$

Consequently, state updates are computed using a logistic activation function $p(z_k = 1) = [1 + \exp(u_k)]^{-1} =: \sigma(u_k)$.

In a restricted Boltzmann machine (RBM), the units are subdivided into a visible and a hidden layer, with no within-layer connections (Fig. 1a). To enable classification, an additional label layer can be added, which for training purposes can be treated as part of the visible layer. During training, weights and biases are iteratively updated in order to optimize the marginal distribution $p(v, l|h)$ as the underlying distribution for the set of training samples.

Recently, it has been shown how networks of spiking neurons can perform equivalent computations[28], which we briefly outline in the following. The building blocks for our spiking networks are conductance-based leaky integrate-and-fire (LIF) neurons, with membrane potential dynamics governed by

$$C_m \frac{du}{dt} = g_l(E_l - u) + I^{syn}, \qquad (3)$$

where C_m is the membrane capacitance, g_l and E_l the leak conductance and potential, and I^{syn} the synaptic current. Neurons fire upon reaching a threshold voltage, which causes the membrane to be clamped to a reset potential for a duration equal to the refractory period τ_{ref}. The synaptic current is modeled as a sum of exponential kernels triggered by presynaptic spikes s with a synaptic time constant τ_{syn} and weighted by synaptic weights $W_i(t)$ and reversal potentials E_i^{rev}:

1:	Given adaptive weights $\{\mathbf{g}_k\}_{k=1}^{K}$ and the initial configuration of the state z^1 at temperature 1, $k = 1$:
2:	**for** $t = 1{:}T$ (number of iterations) **do**
3:	Given z^t, sample a new state z^{t+1} from $p(z\|k^t)$ by Gibbs sampling.
4:	Given k^t, sample k^{t+1} from proposal distribution $q(k^{t+1} \leftarrow k^t)$. Accept with probability: $\min\left(1, \frac{p(z^{t+1},k^{t+1})q(k^t \leftarrow k^{t+1})g_{k^t}}{p(z^{t+1},k^n)q(k^{t+1} \leftarrow k^t)g_{k^{t+1}}}\right)$
5:	Update adaptive adjusting factors: $g_i^{t+1} = g_i^t(1 + \gamma_t I(k^{t+1} = i)), \quad i = 1, \ldots, K.$
6:	**end for**
7:	Collect data: Obtain (dependent) samples from target distribution $p(z)$ by keeping $k = 1$.

Table 1. Adaptive simulated tempering.

$$I^{\text{syn}}(t) = \sum_s \sum_i W_i(t)(E_i^{\text{rev}} - u)\exp[-(t - t^s)/\tau_{\text{syn}}]. \tag{4}$$

The temporal dependence of the synaptic weights accounts for the STP mechanism we discuss later.

Each neuron receives both functional synaptic input from other neurons within the network and diffuse background input from external neurons that can be modeled as Poisson spike trains. The latter causes the neuron to fire stochastically. Since, at the level of spikes, the output of a neuron can be considered binary, we associate a binary random variable z_k to each neuron. As a neuron never fires within the refractory period, it is natural to set $z_k = 1$ for $t \in [t_k^s, t_k^s + \tau_{\text{ref}})$ and 0 otherwise (Fig. 1b).

For constant functional synaptic input as defined above, the mean firing rate of such a neuron is proportional to its activation function $p(z_k = 1)$. By applying strong background input, we lift neurons into a high-conductance state (HCS)[29], which molds their activation function into an approximately logistic shape[30]:

$$p(z_k = 1) \approx \sigma\,(\alpha u_k^{\text{f}} - \beta), \tag{5}$$

with scaling parameters α and β, where u_k^{f} represents the functional, i.e., background-free, membrane potential. Similarly to Gibbs sampling, the functional membrane potential thereby fulfills the local computability condition (equation 2), which is a sufficient computational prerequisite for sampling in neural networks[23,31]. The scaling parameters can be derived analytically and allow a direct translation of the BM parameters W and b to the corresponding parameters in the biological domain (Methods, Sec. 1).

Training. To speed up training, we used RBMs with binary units. As a learning algorithm, we used the coupled adaptive simulated tempering (CAST) method[16], which is a version of the wake-sleep algorithm. In CAST, two instances of the RBM are simulated in parallel, with one of them staying at a constant inverse temperature $\beta = 1$ for parameter update using persistent contrastive divergence[32] (slow chain) and the other one using adaptive simulated tempering (AST)[16] for mixing (fast chain). The states of the two RBMs are swapped constantly to help the slow chain jump out of local minima during parameter updating. In AST, states $z^{(t+1)}$ are updated by Gibbs sampling from $p(z|\beta_T^{(t)})$. After each state update, the temperature is itself updated by an adaptive rule that ensures the algorithm spends a roughly equal amount of time at each value β_T. Details of the AST algorithm are described in Table 1. The training hyperparameters for different experiments can be found in Methods Sec. 2.

The trained RBM parameters are then mapped to the spiking-network domain as described below[28] (see Methods Sec. 1 for more details):

$$W_{kj} = \frac{1}{\alpha C_{\text{m}}} \frac{w_{kj}\big(E_{kj}^{\text{rev}} - \mu\big)}{1 - \frac{\tau_{\text{syn}}}{\tau_{\text{eff}}}}\left[\tau_{\text{syn}}(e^{-1} - 1) - \tau_{\text{eff}}\left(e^{-\frac{\tau_{\text{syn}}}{\tau_{\text{eff}}}} - 1\right)\right], \tag{6}$$

where W_{kj} denotes the peak synaptic conductance (see equation 4), C_{m} the membrane capacitance, w_{kj} the abstract Boltzmann weight, E_{kj}^{rev} the corresponding reversal potential, μ the mean free membrane potential, τ_{syn} the synaptic time constant and $\tau_{\text{eff}} = C_{\text{m}}/\langle g_{\text{tot}}\rangle$ the (mean) effective membrane time constant.

Tempering vs. short-term plasticity. When trained from data, the energy landscape $E(z)$ is shaped in a way that assigns low energy values (modes) to the samples in the training data. If this dataset is composed of very dissimilar classes, training algorithms tend to separate them by high energy barriers[16,33]. As their height grows during training, Gibbs sampling becomes increasingly ineffective at covering the entire relevant state space, as reflected by a high correlation between consecutive samples caused by the component-wise update of states[16,17,33,34]. Consequently, a BM would need longer to converge towards its underlying distribution. This problem becomes particularly inconvenient when dealing with complex, real-world data, or when an agent must rely on the prediction of the network to make a fast decision.

The ability of a sampling-based generative model to jump across energy barriers, also known as mixing, has therefore received significant attention[16,17,35,36]. Many of these methods rely on some version of simulated

tempering, which modifies the temperature parameter β_T in order to globally flatten the network's energy landscape (Fig. 1c). Therefore, in addition to conventional Gibbs sampling, we use the AST algorithm[16] as a benchmark for our spiking networks (Methods, Sec. 2).

While greatly increasing the mixing capabilities of generative networks, it is important to note that all tempering schedules come with a cost of their own, both because they require additional computations and because they only gather valid samples at low temperatures ($\beta \approx 1$), thereby effectively slowing down the sampling process. Furthermore, they require parameter changes that assume knowledge about the global state of the network, which is difficult to reconcile with biology. This motivates the search for a local update rule that has biological relevance, improves mixing and can be embedded in spiking networks.

In biological neural networks, the momentary synaptic interaction strength is reflected in the size of the elicited postsynaptic potential (PSP). In dynamic synapses, this value may change over time depending on the presynaptic activity. To model this dependence, we use the Tsodyks-Markram model of STP[37]:

$$PSP \propto w \cdot U \cdot R \tag{7}$$

$$dR/dt = (1 - R)/\tau_{\mathrm{rec}} - U \cdot R \cdot \delta(t - t_s) \tag{8}$$

$$dU/dt = -U/\tau_{\mathrm{fac}} + U_0 \cdot (1 - U) \cdot \delta(t - t_s). \tag{9}$$

Here, w represents the (static) synaptic weight and $U \in [0, 1]$ the utilized fraction of available synaptic resources $R \in [0, 1]$. Upon arrival of a presynaptic spike at time t_s, the synapse is depressed by subtracting U from R, which recovers exponentially with the time constant τ_{rec}. Facilitation is modeled by a simultaneous increase in U, followed by an exponential decay with time constant τ_{fac}.

Since both tempering and STP effectively modify the energy landscape by changing network parameters during sampling, they clearly bear some conceptual resemblance. However, while tempering simultaneously affects all synaptic weights, STP only affects the efferent connections of those neurons that are simultaneously active at a given moment in time. Therefore, in contrast to the global modifications of the energy landscape incurred by tempering, STP has a more local effect, as sketched in Fig. 1c. Note that this effect is not equivalent to neuronal adaptation, because it does not prohibit neurons from remaining active over extended periods, which is essential for generating consecutive patterns with significant overlap.

Results

We study the effects of STP on the performance of spiking networks trained for different tasks. We start by discussing how STP can improve the sampling accuracy of small networks configured to sample from a fully specified target distribution, even when the energy landscape is shallow enough to not cause mixing problems. This is no longer the case for hierarchical networks trained directly on data, for which we study the influence of STP on both their generative and their discriminative properties. Furthermore, we show how STP can aid pattern completion in a network trained on a highly imbalanced dataset. For all spiking network simulations, we used NEST[38] with PyNN[39] as a front-end.
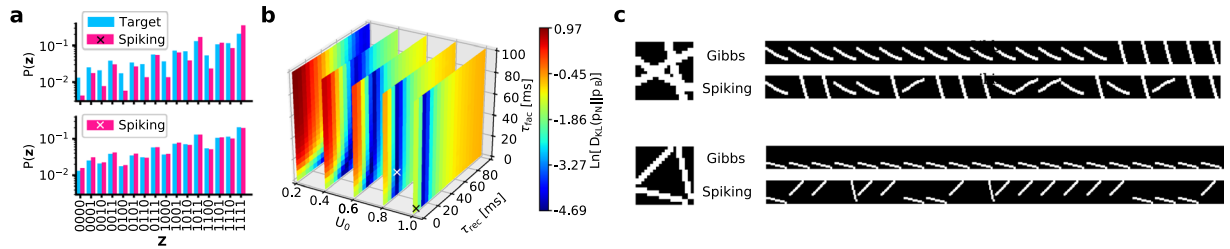
Sampling from a fully specified target distribution. By modulating synaptic interactions, STP shapes the sampled distribution. This can be helpful when a spiking network needs to approximate a distribution that is otherwise incompatible with biological neuro-synaptic dynamics, as we discuss in the following.

Consider the case where the target distribution of the spiking network is a Boltzmann distribution. When a neuron needs to continuously represent a state $z_k(t) = 1$ for an extended period, it fires a sequence of n spikes at maximum frequency $1/\tau_{\mathrm{ref}}$. Following equation 2, the resulting PSPs should increase a postsynaptic neuron's membrane by a constant $\Delta u_i = w_{ik}$, which implies a rectangular PSP shape. However, this is not a realistic shape for a more biologically plausible scenario, where PSPs have an exponentially shaped decay. This causes them to accumulate (Fig. 1d), such that the average increment $\langle \Delta u_i \rangle_n$ becomes a function of the burst length n, thereby distorting the sampled distribution.

Synaptic depression can mitigate this effect (Fig. 2a) by causing a gradual decrease in the amplitude of consecutive PSPs. Indeed, when sweeping over the $(U_0, \tau_{\mathrm{rec}}, \tau_{\mathrm{fac}})$ parameter space (Fig. 2b), we find that an optimal reproduction of the target distribution is achieved for $\tau_{\mathrm{rec}} \approx 15$ ms, which is close to the synaptic time constant of $\tau_{\mathrm{syn}} = 10$ ms. This affords an intuitive explanation: In the HCS, the effective membrane time constant becomes small and τ_{syn} dominates the PSP decay. If the recovery of synaptic resources R (equation 8) happens at the same speed as the PSP decay, the STP mechanism essentially emulates a renewing synapse with an approximately constant running average (Fig. 1d). The slightly larger optimal recovery time constant further compensates for the long tails of exponential PSPs, which potentiate interaction strengths compared to the ideal case of rectangular PSPs (Methods, Sec. 1). Note that the manifold for which the target distribution is close-to-optimally reproduced contains many different STP configurations, including the range of biologically observed parameters[40], but not the $(u, \tau_{\mathrm{rec}}, \tau_{\mathrm{fac}}) = (1, 0, 0)$ triplet for static synapses (Fig. 2b).

For the example in Fig. 2a, we used a fully specified target distribution $p_B(z|W, b)$; training was not needed, as synaptic weights can be computed directly from the parameters W and b (Eqn. 10). Here, we used a target Boltzmann distribution with randomly drawn parameters that produce a diverse energy landscape, but not so rough as to create problems with mixing. This changes when the network parameters are learned from data, as we discuss in the following.

Mixing in a simple learning scenario. Borrowing from observations in the early visual system, we generated images of oriented bars. The bars were positioned in a way that gave rise to an "easy" (overlapping) and a

Figure 2. Sampling from a fully specified target distribution and mixing in a simple learning scenario. (**a**) Distribution sampled by the spiking network for two different configurations of synaptic parameters. Depressing synapses (bottom) allow the network to come much closer to the target distribution (blue) than non-plastic ones (top). (**b**) Kullback-Leibler divergence between sampled (p_N) and target (p_B) distribution of a spiking network with 10 neurons (5 hidden, 5 visible) for different STP parameters ($U_0, \tau_{rec}, \tau_{fac}$). Note that many different parameter combinations lead to close to optimal (white cross) sampling, but static synapses (black cross) are not among them. (**c**) Left: Training data for the easy (top) and hard (bottom) learning scenario. The 3 images from the training set, each containing a single oriented bar, are superimposed to highlight the overlap of the oriented bars (or lack thereof). Right: Sequence of images generated by a Gibbs sampler and an STP-endowed spiking network with equivalent parameters W and b. For each method, 20 samples are taken from 5000 consecutively generated images with equal interval.

"hard" (non-overlapping) dataset (Fig. 2c). We then trained a two-layer hierarchical network (400 visible, 30 hidden units) on each of these datasets using CAST[16]. Intuitively, the difficulty of learning a generative model of this data increases when the bars have little or no overlap: in this case, training gives rise to three nearly disjoint populations that have, on average, excitatory connections within and inhibitory connections between them. The emergence of such a population-based winner-take-all structure can be characterized by the mean interaction strength $\overline{w}_{ij} = \langle z_i^T \rangle W \langle z_j \rangle$ between two population activity vectors $\langle z_i \rangle$ and $\langle z_j \rangle$, which represent the average network activity during the presentation of the ith and jth input pattern, respectively. For the easy dataset, learning gave rise to a mean within-population interaction strength of $\langle \overline{w}_{ii} \rangle_i = 92.75$ and a mean between-population interaction strength of $\langle \overline{w}_{ij} \rangle_{i \neq j} = -145.48$. These values changed to $\langle \overline{w}_{ii} \rangle_i = 102.82$ and $\langle \overline{w}_{ij} \rangle_{i \neq j} = -164.66$ for the hard dataset, reflecting the increased competition and disjointedness between the three emerging populations. STP, however, can weaken active synapses, temporarily reducing $|\langle \overline{w} \rangle|$ to enable switching between attractors.

The learned parameter set was used to compare the performance of classical Gibbs sampling and STP-endowed spiking networks (Fig. 2c). For the easy dataset, both the Gibbs sampler and the spiking network were able to mix, although the former spent on average 100 times longer in the same mode before switching, thereby requiring more time to converge to the target distribution. For the hard dataset, the spiking networks retained their ability to mix, whereas Gibbs sampling was unable to leave the (randomly initialized) local mode. These observations mirror those found in studies of cortical attractor networks[41]. While this simple experimental setup was specifically designed to illustrate the potential problems of sampling-based generative models and the ability of STP-endowed spiking networks to circumvent them, we show in the following that these properties are preserved in more complex scenarios.

Generation and classification of handwritten digits. The problem of mixing becomes even more pronounced when dealing with larger, more complex datasets. Here, we trained a hierarchical 3-layer network with 784 visible, 600 hidden and 10 label units on handwritten digits from the MNIST dataset[42]. By treating the label units as part of the visible layer during training, we simultaneously trained a generative and a discriminative model of the data. This objective is particularly challenging, because mechanisms that improve mixing tend to disrupt classification and vice-versa[17].

To evaluate the quality of generated samples, we computed a log-likelihood estimation of 2000 test images (not used during training) using the indirect sampling likelihood (ISL) method[33,34] (see also Methods). Due to the size of the network, a full scan of the parameter space for finding optimal STP parameters was no longer feasible. Therefore, starting from a good parameter set found by trial and error, we performed two 2D-scans of the ($U_0 \tau_{rec}, \tau_{fac}$) parameter space (Fig. 3a). As in the previous examples, we found short-term depression to be essential for achieving high ISL values. Furthermore, a small value of U_0 combined with short-term facilitation was also beneficial, allowing an initial strengthening followed by a weakening of the active attractor, as sketched in Fig. 1c,d. Similar observations have been made in cortex, where STP can promote the enhancement of transients[43].

We used one of the optimal STP parameter sets ($U_0 = 0.01$, $\tau_{rec} = 280$ ms) to compare the generative performance of spiking networks to classical Gibbs sampling. Due to its improved mixing capability, the spiking network was able to quickly cover a large portion of the relevant state space, as reflected by a faster ISL gain during sampling (Fig. 3b). This is a systematic effect and only weakly dependent on initial conditions, as can be seen in Fig. 3c, which shows a histogram over 100 random seeds. For this comparison, we chose a sampling duration of 10 s as a conservative estimate for the maximum duration for a biological agent to experience stable stimulus conditions and therefore sample from a stable target distribution. The faster mixing is the result of the spiking network's ability to jump out of local attractors, which is reflected in a much shorter time spent on average within
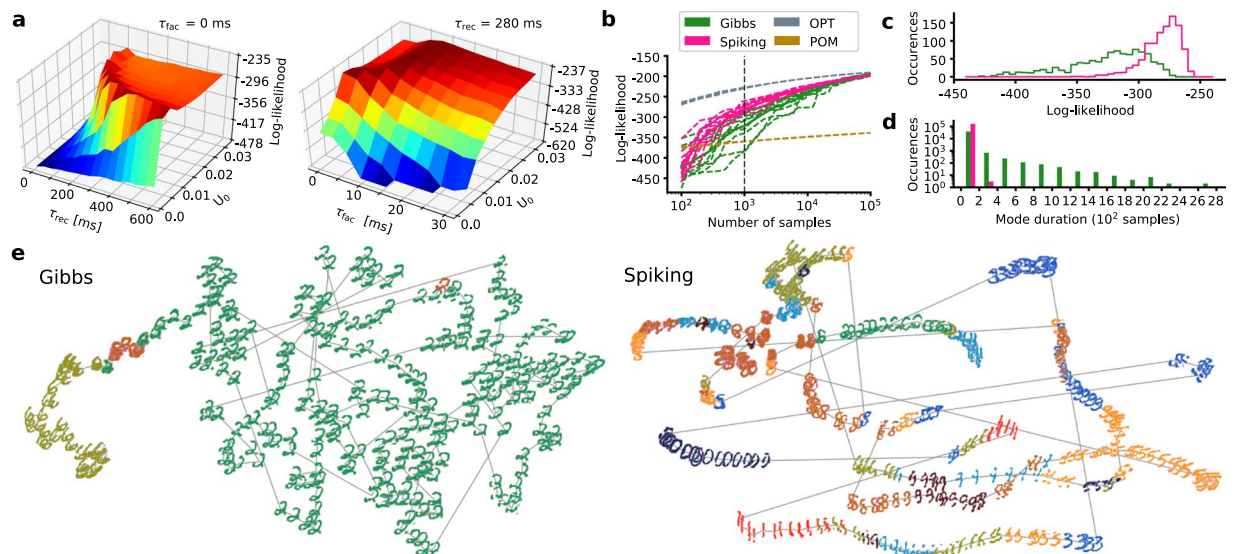
Figure 3. Superior generative performance of an STP-endowed spiking network compared to an equivalent Gibbs sampler. (**a**) 2D parameter scans of the STP parameters (U_0, τ_{rec}, τ_{fac}) with multiple configurations leading to good generative performance. (**b**) Log-likelihood from ISL of the test set calculated from an increasing number of samples. Each sampling method was simulated with 10 different random seeds (dashed lines) and their mean value was calculated (solid lines). The ISLs of an optimal sampler with the same parameters (OPT, gray) and the product of marginals (POM, brown) are shown for comparison (see Methods Sec. 3). (**c**) Direct comparison between the two sampling methods for 10^3 samples, equivalent to a sampling duration of 10 s in the biological domain. ISL histogram generated from 100 random seeds. (**d**) Histogram of times spent within the same mode when no visible units are clamped. (**e**) tSNE plots of images produced by the two methods over 1800 consecutive samples. For every 6th of these samples, an output image is shown. Consecutive images are connected by gray lines. Different colors represent different image classes, defined by the label unit that showed the highest activity at the time the sample was generated. Note that tSNE inherently normalizes the area of the 2D projection; the volume of phase space covered by the Gibbs chain is, in fact, much smaller than the one covered by the spiking network.

the same mode (Fig. 3d). Here, we defined a mode as the dominant class of the currently represented image; a mode was therefore defined by the identity of the neuron in the label layer with the highest firing rate.

It is important to note that, due to the STP-modulated interaction, the spiking network does not sample from the exact same distribution as the Gibbs sampler, despite using an equivalent (W, b) parameter set. However, for a very large number of samples ($>10^5$), the two methods converge towards the same ISL (Fig. 3b), indicating that the discrepancy in performance for shorter sampling durations is not due to a fundamental difference in their respective ground truths.

While the ISL, as an abstract quantity, provides a useful numerical gauge of the quality of a generative model, a direct depiction of the produced images is particularly instructive. Here, we used the t-distributed stochastic neighbor embedding (tSNE) method[44] (see also Methods) for a 2D-embedding of the high-dimensional sampled distribution. The similarity between samples is largely reflected by their 2D distance and a large jump can be interpreted as a switch between attractors. As seen in Fig. 3e, the spiking network produces a significantly more diverse set of samples compared to the Gibbs sampler.

When the visible layer is clamped to a particular input, the same network can be used as a discriminative model of the learned data. Using the same parameters as for the generative task, the benchmark Gibbs sampler obtained a classification accuracy of 93.4% on the MNIST test data. The spiking network with STP performed only slightly worse, at 93.2%. The additional generative capabilites gained by the spiking networks through STP were therefore not strongly detrimental to their classification accuracy.

Modeling imbalanced datasets.
In many real-world scenarios, the available data is imbalanced, with much of the data belonging to one class and significantly less samples being distributed over others. It is well-known that imbalanced data can cause severe problems for data mining and classification[45,46]. One solution is to create a more balanced dataset from the imbalanced one, which can be achieved by methods such as under- or over-sampling[46,47]. However, such an a-priori modification of the input data does not seem biologically plausible. Still, cognitive biological agents appear to easily overcome this problem: humans will have little difficulty imagining a platypus from seeing only its bill, despite having likely seen many more ducks throughout their lifetime. Spiking networks with STP provide a simple solution to the problem of imbalanced training data, without any need for preprocessing.

We generated an imbalanced dataset of 1000 images by randomly selecting 820 digits of class "1" and 45 from the "0", "2", "3" and "8" classes. After training, we compared the generative output of a Gibbs sampler, an AST sampler and a spiking network with STP. Note that the effective sampling speed of AST is roughly 20 times slower
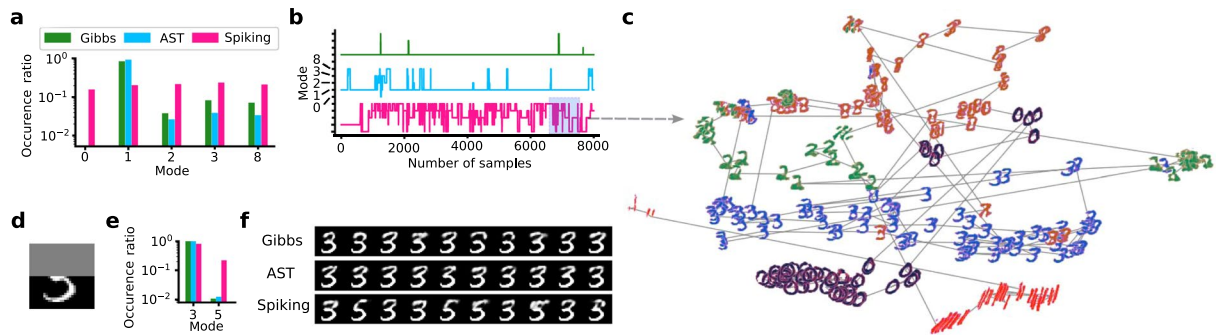
Figure 4. Comparison of Gibbs and AST samplers with STP-endowed spiking networks for imbalanced training data. (**a**) Histogram of relative time spent in different modes calculated from 16,000 samples. (**b**) Mode evolution over 8,000 consecutive samples. (**c**) tSNE plot of images generated by the spiking network over a duration of 10 s, with 40 ms between consecutive images. (**d**) Ambiguous input to the visible layer. The upper half is not clamped and free to complete the pattern. (**e**) Histogram of relative time spent in different modes during the pattern completion task, measured over 20,000 consecutive samples. (**f**) Comparison of the sequence of images generated by the different methods over 5000 samples (only every 500th is shown).

compared to Gibbs sampling, since most of the produced samples are not considered valid. In this scenario, it becomes particularly useful that the spiking network transiently modifies the learned data distribution (Fig. 4a). The STP-induced weakening of active attractors balances out their activity, thereby negating the inherent imbalance induced by the training data. Furthermore, as observed before, the spiking network switches faster between modes (Fig. 4b,c).

These abilities become particularly useful in a scenario of inference based on incomplete information, for which pattern completion is a prime example. Here, we used a training set with 6 majority classes ("0", "1", "2", "3", "4", "6", 800 samples each) and one minority class (200 samples of "5"). We generated an ambiguous image by clamping the lower half of the visible layer to a configuration compatible with both a "3" and a "5" (Fig. 4d). While Gibbs and AST strongly undersample the minority class, the spiking network produces a much more balanced set of images, with swift transitions between modes (Fig. 4e,f). The estimate of the possible realities underlying the incomplete observation is therefore improved both on long and on short time scales. This can be particularly useful for an agent in need of a quick reaction, as, for example, often required in nature in a fight-or-flight scenario.

Discussion

We have shown how a combination of spike-based communication and short-term plasticity can enhance the ability of neural networks to perform probabilistic inference in high-dimensional data spaces. Here, a spike-triggered plasticity rule played a similar role to simulated tempering methods used for classical neural networks, but without requiring complex computations on the global network state or long waiting times between valid samples. The spiking networks outperformed their classical counterparts as generative models of real-world data, with little disturbance to their classification capability, which we expect to be largely remediable by additional fine-tuning of the network parameters. Furthermore, they were also able to cope with imbalanced training data, as demonstrated by their superior performance in a pattern completion task on ambiguous input. Intriguingly, the synaptic parameters used to achieve this performance are compatible to experimental data[40].

As a potential downside of the functional gains discussed in the manuscript, the inclusion of more complex membrane and synapse dynamics are likely to increase the computational cost of applying our paradigm to classical neural networks such as Boltzmann machines. However, we expect a simple, local synaptic update rule to be overall more efficient than global updates required by, e.g, tempering schedules. Furthermore, the increased performance should be largely independent of the particular shape of a PSP and thereby of the used neuron model. Moreover, in physical neuromorphic emulation, added complexity in neural dynamics incurs no runtime penalty[21,22].

Depending on the nature of the sampled distribution and the associated optimal parameters, STP can play a dual role. For low-dimensional spaces in which networks only rarely have mixing problems, STP can narrow the gap between the sampled and the target distribution. On the other hand, in large networks trained from data, both the improved mixing and the balancing effect represent functionally advantageous distortions of the network's underlying Boltzmann distribution. The deviations are a consequence of both the dynamic nature of plastic weights as well as due to emerging asymmetries in the connectivity matrix.

The STP parameters themselves require only little tuning, as evidenced by the comparatively large volume in parameter space that enhances performance, especially for high-dimensional problems. However, the optimal parameter set may vary, depending on the nature of the learned problem. In a machine learning context, various algorithms for meta-parameter optimization have been proposed and could be applied to STP as well[48–50]. With respect to biology, as the function and location of individual brain areas remains largely conserved both within and among species, we speculate evolution to have played a key role in parameter optimization.

In fact, it was suggested that during a working memory task studied *in vivo* with rats, short-term synaptic depression in the medial prefrontal cortex sets the "life time" of high-dimensional neuronal assemblies that code for the integrated representation of position and sensory inputs[51]. While the rat navigates in a maze, the

	COBA	CUBA	
C_{m}	0.1 nF	0.2 nF	membrane capacitance
τ_{m}	20 ms	0.1 ms	membrane time constant
τ_{ref}	10 ms	10 ms	refractory time constant
τ_{syn}	10 ms	10 ms	synaptic time constant
ϑ	−50 mV	−50 mV	threshold voltage
ρ	−53 mV	−50.01 mV	reset potential
$E_{\mathrm{exc}}^{\mathrm{rev}}$	0 mV	—	excitatory reversal potential
$E_{\mathrm{inh}}^{\mathrm{rev}}$	100 mV	—	inhibitory reversal potential

Table 2. Neuron parameters.

representation moves from one assembly to another on a time scale that roughly corresponds to the one of synaptic depression. Short-term synaptic plasticity, originally found in rat sensory cortices[52], has also been found in the prefrontal cortex using paired recordings *in vitro*[53].

In a physical system such as a biological brain, the studied plasticity mechanism essentially comes for free, as it only requires a limited pool of synaptic resources. Together with other activity-modulating mechanisms such as neuronal adaptation, it could be a key contributor to the ability of the brain to navigate efficiently in a very-high-dimensional stimulus space. Importantly, these mechanisms provide immediate computational advantages for spike-based neuromorphic devices, facilitating the development of efficient artificial agents that replicate the inferential capabilities of their biological archetypes.

Methods

Spiking networks. To generate our spiking sampling networks, we follow[28]. We emulate an HCS by stimulating the LIF neurons (Conductance-based (COBA), Table 2) with balanced excitatory and inhibitory Poisson noise This produces an approximately logistic activation function (Fig. 5a), parametrized by a shift β and a scaling parameter α (equation 5). These parameters can be used to translate synaptic interaction strengths from the Boltzmann domain to synaptic conductances:

$$W_{kj} = \frac{1}{\alpha C_{\mathrm{m}}} \frac{w_{kj}\big(E_{kj}^{\mathrm{rev}} - \mu\big)}{1 - \frac{\tau_{\mathrm{syn}}}{\tau_{\mathrm{eff}}}} \bigg[\tau_{\mathrm{syn}}(e^{-1} - 1) - \tau_{\mathrm{eff}}\bigg(e^{-\frac{\tau_{\mathrm{syn}}}{\tau_{\mathrm{eff}}}} - 1\bigg)\bigg],$$

(10)

where W_{kj} denotes the peak synaptic conductance (see equation 4), C_{m} the membrane capacitance, w_{kj} the abstract Boltzmann weight, E_{kj}^{rev} the corresponding reversal potential, μ the mean free membrane potential, τ_{syn} the synaptic time constant and $\tau_{\mathrm{eff}} = C_{\mathrm{m}}/\langle g_{\mathrm{tot}}\rangle$ the (mean) effective membrane time constant. This corresponds to a match of the average interaction during the refractory period of the presynaptic neuron (Fig. 5b). This setup allows an accurate sampling from target Boltzmann distributions (Fig. 5c,d).

To speed up simulations, we used an effective current-based (CUBA) model to replace the COBA one (Table 2). Figure 5e shows a comparison between the two models. Under appropriate parametrization, we could reduce the background input rates from $\nu = 5$ kHz to $\nu = 0.4$ kHz.

Training hyperparameters. The used hyperparameters (number of epochs T, batch size N, learning rate η) were based on suggestions from previous work[54] and empirical experience. For all datasets, we used 20 equidistant inverse temperatures $\beta_k \in [0.9, 1]$. The adaptive weights $\{\mathbf{g}_k\}_{k=1}^{K}$ were initialized to 1 for all temperatures and as $\gamma_t \to 0$ the adaptive weights will converge. In all experiments, we set γ_t as $90/(150 + t)$. For the bar example (Fig. 2), we used $T = 100{,}000$, $N = 3$ and $\eta = 10/(2000 + t)$. For the full MNIST example (Fig. 3), we used $T = 200{,}000$, $N = 100$ and $\eta = 40/(t + 2000)$. For the first example of an imbalanced dataset (Fig. 4a–c), we used a network with 784 visible, 10 label and 400 hidden units with $T = 100{,}000$, $N = 100$ and $\eta = 20/(t + 2000)$. For the example of pattern completion from an imbalanced dataset (Fig. 4d,e), we used a network with 784 visible, 10 label and 400 hidden units with $T = 200{,}000$, $N = 100$ and $\eta = 40/(t + 2000)$.

Indirect sampling likelihood. To have a quantitative comparison of mixing between different sampling procedures, we used the indirect sampling likelihood (ISL) method[33,34]. The method constructs a non-parametric density estimator to evaluate how close each test example is from any of the generated examples. The likelihood of a test sample $\mathbf{y}$ given a series of generated sample $\{\mathbf{x}_i\}$ is defined as:

$$p(y) = \frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{d} \beta^{1_{y_j = x_{ij}}}(1 - \beta)^{1_{y_j \neq x_{ij}}},$$

(11)

where N is the number of generated samples, d is the dimension of y or x_i and β is a hyperparameter which controls the gain (β) and punishment ($1 - \beta$) to the likelihood when comparing the test sample with the generated sample. We set $\beta = 0.95$ to optimize the likelihood; other values ($\beta \in (0.5, 1]$) would rescale the likelihood but without causing qualitative differences.

In Fig. 3b, we plot the mean log-likelihood of 2000 samples from the test set against the number of generated samples. The faster increase of the ISL curve for the spiking network is due to better mixing, as the generated
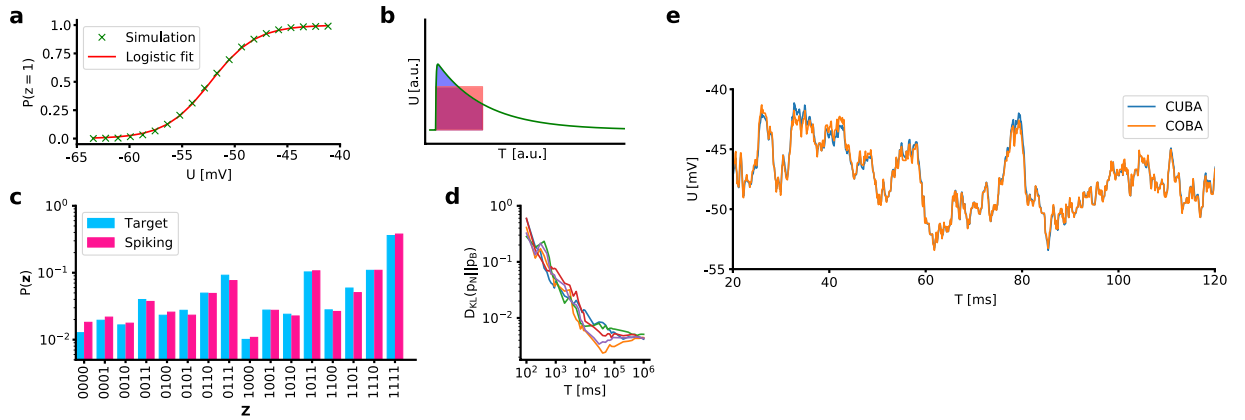
Figure 5. Spiking sampling networks. (**a**) Activation function of an LIF neuron in the HCS and logistic fit. (**b**) Sketch of synaptic weight translation (equation 10). (**c**) Sampled distribution of a fully connected 4-neuron LIF network vs. target distribution. (**d**) Evolution of Kullback-Leibler divergence between sampled (p_N) and target (p_B) distribution for 5 different random seeds. (**e**) Free membrane potential ($\vartheta = 0$) of a biologically plausible COBA LIF neuron in the HCS compared to an equivalent CUBA LIF neuron (parameters given in Table 2).

samples cover the main modes of the test samples faster (Fig. 3d,e). To provide a frame of reference, we also plotted two additional ISL curves. The POM (product of marginals) sampler generated images by sampling each pixel individually from its intensity distribution over the entire training set. This sampler preserves the marginal probability distributions for each pixel, but discards any further structure of the image (encoded in correlations between pixel intensities). The OPT (optimal) sampler started out with a base set of 10^5 images generated with AST, from which it randomly picked images sequentially. This guarantees optimal mixing for the underlying model, because the base set covers all main modes of the state space, but consecutive samples have no correlation.

t-distributed stochastic neighbor embedding. The tSNE method[44] finds a low-dimensional map for a high-dimensional data set, in which the similarity between samples is reflected by their distances in the low-dimensional map. Here, we projected the generated digits to a plane to provide an intuitive understanding of the network dynamics and the mixing between different modes (digit classes).

The Euclidean distances between high-dimensional samples $\{x_i\}$ are converted into symmetric pairwise similarities

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n},$$

(12)

where n is the number of samples and $p_{j|i}$ is a conditional probability:

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)},$$

(13)

with variance σ_i, which is determined by first defining a so-called perplexity value as the effective number of neighbors of a data point, and then running a binary search. For the low-dimensional points y_i and y_j mapped from the high-dimensional data points $\mathbf{x}_i$ and $\mathbf{x}_j$, the similarity is defined using a t-distribution with one degree of freedom:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_k - y_l\|^2)^{-1}}.$$

(14)

If the mapped points correctly model the similarity between the high-dimensional data points, the similarities p_{ij} and q_{ij} will be equal.

With this motivation, tSNE minimizes the sum of Kullback-Leibler divergences over all data points using a gradient descent method. The cost function C is given by

$$C = D_{KL}(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

(15)

Its gradient with respect to the map point i can then be derived to provide an update of the mapping:

$$\Delta y_i \propto \frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1}.$$

(16)

References

1. McCulloch, W. S. & Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* **5**, 115–133 (1943).
2. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review* **65**, 386 (1958).
3. Crick, F. The recent excitement about neural networks. *Nature* **337**, 129–132 (1989).
4. Stork, D. G. Is backpropagation biologically plausible. In *International Joint Conference on Neural Networks*, vol. 2, 241–246 (IEEE Washington, DC, 1989).
5. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
6. Lillicrap, T. P., Cownden, D., Tweed, D. B. & Akerman, C. J. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications* **7** (2016).
7. Lee, J. H., Delbruck, T. & Pfeiffer, M. Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience* **10** (2016).
8. Neftci, E. O., Augustine, C., Paul, S. & Detorakis, G. Event-driven random back-propagation: Enabling neuromorphic deep learning machines. *Frontiers in Neuroscience* **11** (2017).
9. Petrovici, M. A. *et al.* Pattern representation and recognition with accelerated analog neuromorphic systems. *Proceedings of the 2017 IEEE International Symposium on Circuits and Systems* https://arxiv.org/abs/1703.06043 (2017).
10. Zipser, D. & Andersen, R. A. A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature* **331**, 679–684 (1988).
11. Kriegeskorte, N. Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annual Review of Vision Science* **1**, 417–446 (2015).
12. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural networks* **61**, 85–117 (2015).
13. Hinton, G. *et al.* Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* **29**, 82–97 (2012).
14. Goodfellow, I. *et al.* Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680 (2014).
15. Desjardins, G., Courville, A., Bengio, Y., Vincent, P. & Delalleau, O. Parallel tempering for training of restricted boltzmann machines. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 145–152 (MIT Press Cambridge, MA, 2010).
16. Salakhutdinov, R. Learning deep boltzmann machines using adaptive mcmc. In *Proceedings of the 27th International Conference on Machine Learning* (ICML-10), 943–950 (2010).
17. Bengio, Y., Mesnil, G., Dauphin, Y. & Rifai, S. Better mixing via deep representations. In *ICML* (*1*), 552–560 (2013).
18. Fiser, J., Berkes, P., Orbán, G. & Lengyel, M. Statistically optimal perception and learning: from behavior to neural representations. *Trends in cognitive sciences* **14**, 119–130 (2010).
19. Jezek, K., Henriksen, E. J., Treves, A., Moser, E. I. & Moser, M.-B. Theta-paced flickering between place-cell maps in the hippocampus. *Nature* **478**, 246 (2011).
20. Hindy, N. C., Ng, F. Y. & Turk-Browne, N. B. Linking pattern completion in the hippocampus to predictive coding in visual cortex. *Nature neuroscience* **19**, 665 (2016).
21. Pfeil, T. *et al.* Six networks on a universal neuromorphic computing substrate. *Frontiers in neuroscience* **7** (2013).
22. Schemmel, J. *et al.* A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *Circuits and systems (ISCAS), proceedings of 2010 IEEE international symposium on*, 1947–1950 (IEEE, 2010).
23. Smolensky, P. Information processing in dynamical systems: Foundations of harmony theory. Tech. Rep., DTIC Document (1986).
24. Larochelle, H. & Bengio, Y. Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, 536–543 (ACM, 2008).
25. Salakhutdinov, R. & Hinton, G. E. Deep boltzmann machines. In *AISTATS* vol. 1, 3 (2009).
26. Dahl, G. *et al.* Phone recognition with the mean-covariance restricted boltzmann machine. In *Advances in neural information processing systems*, 469–477 (2010).
27. Srivastava, N. & Salakhutdinov, R. R. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, 2222–2230 (2012).
28. Petrovici, M. A., Bill, J., Bytschok, I., Schemmel, J. & Meier, K. Stochastic inference with spiking neurons in the high-conductance state. *Physical Review E* **94**, 042312 (2016).
29. Destexhe, A., Rudolph, M. & Pare, D. The high-conductance state of neocortical neurons *in vivo*. *Nature Reviews Neuroscience* **4**, 739–751 (2003).
30. Petrovici, M. A., Bytschok, I., Bill, J., Schemmel, J. & Meier, K. The high-conductance state enables neural sampling in networks of lif neurons. *BMC Neuroscience* **16**, O2 (2015).
31. Buesing, L. *et al.* Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons. *PLoS Comput Biol* **7**, e1002211 (2011).
32. Tieleman, T. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, 1064–1071 (ACM, 2008).
33. Breuleux, O., Bengio, Y. & Vincent, P. Unlearning for better mixing. *Universite de Montreal/DIRO* (2010).
34. Desjardins, G., Courville, A., Bengio, Y., Vincent, P. & Delalleau, O. Tempered markov chain monte carlo for training of restricted boltzmann machines. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, vol. 9, 145–152 (2010).
35. Marinari, E. & Parisi, G. Simulated tempering: a new monte carlo scheme. *EPL (Europhysics Letters)* **19**, 451 (1992).
36. Wang, F. & Landau, D. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical review letters* **86**, 2050 (2001).
37. Tsodyks, M., Pawelzik, K. & Markram, H. Neural networks with dynamic synapses. *Neural computation* **10**, 821–835 (1998).
38. Diesmann, M. & Gewaltig, M.-O. Nest: An environment for neural systems simulations. *Forschung und wisschenschaftliches Rechnen, Beiträge zum Heinz-Billing-Preis* **58**, 43–70 (2001).
39. Davison, A. P. *et al.* Pynn: a common interface for neuronal network simulators. *Frontiers in neuroinformatics* **2** (2008).
40. Wang, Y. *et al.* Heterogeneity in the pyramidal network of the medial prefrontal cortex. *Nature neuroscience* **9** (2006).
41. Lundqvist, M., Rehn, M., Djurfeldt, M. & Lansner, A. Attractor dynamics in a modular network model of neocortex. *Network: Computation in Neural Systems* **17**, 253–276 (2006).
42. LeCun, Y. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/ (1998).
43. Abbott, L. & Regehr, W. G. Synaptic computation. *Nature* **431**, 796 (2004).
44. Maaten, Lvd & Hinton, G. Visualizing data using t-sne. *Journal of Machine Learning Research* **9**, 2579–2605 (2008).
45. Chawla, N. V. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*, 853–867 (Springer, 2005).
46. Garca, S. & Herrera, F. Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. *Evolutionary computation* **17**, 275–306 (2009).
47. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research* **16**, 321–357 (2002).

48. Reif, M., Shafait, F. & Dengel, A. Meta-learning for evolutionary parameter optimization of classifiers. *Machine learning* **87**, 357–380 (2012).
49. Thornton, C., Hutter, F., Hoos, H. H. & Leyton-Brown, K. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 847–855 (ACM, 2013).
50. Shahriari, B., Swersky, K., Wang, Z., Adams, R. P. & De Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE* **104**, 148–175 (2016).
51. Fujisawa, S., Amarasingham, A., Harrison, M. T. & Buzsáki, G. Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. *Nature neuroscience* **11**, 823 (2008).
52. Zucker, R. S. & Regehr, W. G. Short-term synaptic plasticity. *Annual review of physiology* **64**, 355–405 (2002).
53. Hempel, C. M., Hartman, K. H., Wang, X.-J., Turrigiano, G. G. & Nelson, S. B. Multiple forms of short-term plasticity at excitatory synapses in rat medial prefrontal cortex. *Journal of neurophysiology* **83**, 3031–3041 (2000).
54. Hinton, G. A practical guide to training restricted boltzmann machines. *Momentum* **9**, 926 (2010).

Acknowledgements

Author Contributions

M.A.P and L.L. conceived and designed the experiments. L.L. performed the simulations. R.M. performed early studies of the generative properties of spiking networks. O.B. developed a software module based on NEST which enabled faster, larger-scale simulations. I.B. was involved in the analysis of results and discussions that shaped the study. L.L. and M.A.P wrote the manuscript. All authors reviewed the manuscript.

Additional Information

Competing Interests: The authors declare no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.