Stochasticity from function – why the Bayesian brain may need no noise

Dominik Dold[a,1,2], Ilja Bytschok[a,1], Akos F. Kungl[a], Andreas Baumbach[a],
Oliver Breitwieser[a], Walter Senn[b], Johannes Schemmel[a], Karlheinz Meier[a], Mihai A. Petrovici[a,b,1,2]

[a] *Kirchhoff-Institute for Physics, Heidelberg University.*
[b] *Department of Physiology, University of Bern.*
[1] *Authors with equal contributions.*
[2] *Corresponding authors*: dodo@kip.uni-heidelberg.de, petrovici@pyl.unibe.ch.

September 24, 2018

Abstract

An increasing body of evidence suggests that the trial-to-trial variability of spiking activity in the brain is not mere noise, but rather the reflection of a sampling-based encoding scheme for probabilistic computing. Since the precise statistical properties of neural activity are important in this context, many models assume an ad-hoc source of well-behaved, explicit noise, either on the input or on the output side of single neuron dynamics, most often assuming an independent Poisson process in either case. However, these assumptions are somewhat problematic: neighboring neurons tend to share receptive fields, rendering both their input and their output correlated; at the same time, neurons are known to behave largely deterministically, as a function of their membrane potential and conductance. We suggest that spiking neural networks may, in fact, have no need for noise to perform sampling-based Bayesian inference. We study analytically the effect of auto- and cross-correlations in functionally Bayesian spiking networks and demonstrate how their effect translates to synaptic interaction strengths, rendering them controllable through synaptic plasticity. This allows even small ensembles of interconnected deterministic spiking networks to simultaneously and co-dependently shape their output activity through learning, enabling them to perform complex Bayesian computation without any need for noise, which we demonstrate in silico, both in classical simulation and in neuromorphic emulation. These results close a gap between the abstract models and the biology of functionally Bayesian spiking networks, effectively reducing the architectural constraints imposed on physical neural substrates required to perform probabilistic computing, be they biological or artificial.

Significance statement

From a generic Bayesian perspective, cortical networks can be viewed as generators of target distributions. To enable such computation, models assume neurons to possess sources of perfect, well-behaved noise - an assumption that is both impractical and at odds with biology. We show how local plasticity in an ensemble of spiking networks allows them to co-shape their activity towards a set of well-defined targets, while reciprocally using the very same activity as a source of (pseudo-)stochasticity. This enables purely deterministic networks to simultaneously learn a variety of tasks, completely removing the need for explicit randomness. While reconciling the sampling hypothesis with the deterministic nature of single neurons, our work also offers an efficient blueprint for in-silico implementations of sampling-based inference.

Introduction

An ubiquitous feature of in-vivo neural responses is their stochastic nature [1–6]. The manifest saliency of this variability has spawned many functional interpretations, with the Bayesian-brain hypothesis arguably being the most notable example [7–12]. Under this assumption, the activity of a neural network is interpreted as representing an underlying (prior) probability distribution, with sensory data providing the evidence needed to constrain this distribution to a (posterior) shape that most accurately represents the possible states of the environment given the limited available knowledge about it.

Neural network models have evolved to reproduce this kind of neuronal response variability by introducing noise-generating mechanisms, be they extrinsic, such as Poisson input [13–16] or fluctuating currents [17–21], or intrinsic, such as stochastic firing [22–27] or membrane fluctuations [28, 29, 19]. However, while representing, to some degree, reasonable approximations, none of the commonly used sources of stochasticity is fully compatible with biological constraints. Contrary to the independent white noise assumption, neuronal inputs are both auto- and cross-correlated to a significant degree [30–36], with obvious consequences for a network's output statistics [37]. At the same time, the assumption of intrinsic neuronal stochasticity is at odds with experimental evidence of neurons being largely deterministic units [38–40]. Therefore, it remains an interesting question how cortical networks that use stochastic activity as a means to perform probabilistic inference can realistically attain such apparent randomness in the first place.

We address this question within the normative framework of sampling-based Bayesian computation [41–45], in which the spiking activity of neurons is interpreted as Markov Chain Monte Carlo sampling from an underlying distribution over a high-dimensional binary state space. We demonstrate how an ensemble of dynamically fully deterministic, but functionally probabilistic networks, can learn a connectivity pattern that enables probabilistic computation with a degree of precision that matches the one attainable with idealized, perfectly stochastic components. The key element of this construction is self-consistency, in that all input activity seen by a neuron is the result of output activity of other neurons that fulfill a functional role in their respective subnetworks. The present work supports probabilistic computation in light of experimental evidence from biology and suggests a resource-efficient implementation of stochastic computing by completely removing the need for any form of explicit noise.

Methods

Neuron model and simulation details

We consider (deterministic) LIF neurons with conductance-based synapses and dynamics described by

$$C_\mathrm{m}\frac{\mathrm{d}u}{\mathrm{d}t} = g_\mathrm{l}\,(E_\mathrm{l}-u) + \sum_{x\in\{\mathrm{e,i}\}} g_x^\mathrm{syn}(E_x^\mathrm{rev}-u)\,, \qquad (1)$$

$$g_x^\mathrm{syn}(t) = \sum_{\mathrm{syn.\,k}}\ \sum_{\mathrm{spikes\ s}} w_k\ \theta(t-t_s)\ \exp\left(-\frac{t-t_s}{\tau^\mathrm{syn}}\right), \qquad (2)$$

$$u(t_s)\geq\vartheta \quad\Rightarrow\quad u(t\in(t_s,t_s+\tau_\mathrm{ref}])=\varrho\,, \qquad (3)$$

with membrane capacitance C_m, leak conductance g_l, leak potential E_l, excitatory and inhibitory reversal potentials $E_\mathrm{e/i}^\mathrm{rev}$, synaptic strength w_k, synaptic time constant τ^syn and firing threshold ϑ. During the refractory period τ_ref, the membrane potential is clamped to the reset potential ϱ. We have chosen the above model because it provides a computationally tractable abstraction of neurosynaptic dynamics [40], but our general conclusions are not restricted to these specific dynamics.

We further use the short-term plasticity mechanism described in [46] to modulate synaptic interaction strengths with an adaptive factor $U_\mathrm{SE}\times R(t)$, where the time-dependence is given by

$$\frac{\mathrm{d}R}{\mathrm{d}t} = \frac{1-R}{\tau_\mathrm{rec}} - U_\mathrm{SE}R\delta(t-t_s)\,,\qquad U_\mathrm{SE}\,,\,R\in[0,1]\,, \qquad (4)$$

with t_s denoting the time of a presynaptic spike and τ_rec the time scale on which the reservoir R recovers. This enables a better control over the inter-neuron interaction, as well as over the mixing properties of our networks [47].

All simulations were performed with PyNN 0.8 [48] and NEST 2.4.2 [49].

Sampling framework

As a model of probabilistic inference in networks of spiking neurons, we adopt the framework introduced in [43, 45]. There, the neuronal output becomes stochastic due to a high-frequency bombardment of excitatory and inhibitory Poisson stimuli (Fig. 1A), elevating neurons into a high-conductance state (HCS), where they attain a high reaction speed due to a reduced membrane time constant. Under these conditions, a neuron's response function becomes approximately logistic and can be represented as $\varphi(u) = (1+\exp(-(u-u_0)/\alpha))^{-1}$ with inverse slope α and inflection point u_0. Together with the mean free membrane potential μ and the mean effective membrane time constant τ_eff, the scaling parameters α and u_0 can be used to translate the weight matrix $\boldsymbol{W}$ and bias vector $\boldsymbol{b}$ of a target Boltzmann distribution $p(\boldsymbol{z})\propto\exp\left(\frac{1}{2}\boldsymbol{z}^T\boldsymbol{W}\boldsymbol{z}+\boldsymbol{z}^T\boldsymbol{b}\right)$ with binary random variables $\boldsymbol{z}\in\{0,1\}^n$ to synaptic weights and leak potentials

Contributions: MAP, DD and IB conceived and designed the study. DD performed the analytical calculations and simulations. OB developed a software module based on NEST and PyNN which enabled faster, larger-scale simulations. AFK provided Python code for setting up SSNs on BrainScaleS. DD and MAP wrote the paper. DD designed and created the figures. All authors reviewed the manuscript.

in a sampling spiking network (SSN):

$$w_{kj} = \frac{\alpha W_{kj} C_\mathrm{m} \frac{\tau_\mathrm{ref}}{\tau^\mathrm{syn}} \left(1 - \frac{\tau^\mathrm{syn}}{\tau_\mathrm{eff}}\right) \left(E^\mathrm{rev}_{kj} - \mu\right)^{-1}}{\left[\tau^\mathrm{syn} \left(e^{-\frac{\tau_\mathrm{ref}}{\tau^\mathrm{syn}}} - 1\right) - \tau_\mathrm{eff} \left(e^{-\frac{\tau_\mathrm{ref}}{\tau_\mathrm{eff}}} - 1\right)\right]} , \tag{5}$$

$$g_\mathrm{l} \boldsymbol{E}_\mathrm{l} = \frac{C_\mathrm{m}}{\tau_\mathrm{eff}} (\alpha \boldsymbol{b} + u_0) - \sum_{x \in \{\mathrm{e},\mathrm{i}\}} \langle g_x^\mathrm{syn} \rangle \, E_x^\mathrm{rev} . \tag{6}$$

This translation effectively enables sampling from $p(\boldsymbol{z})$, where a refractory neuron is considered to represent the state $z_k = 1$ (see Fig. 1B,C).

Measures of network performance

To assess how well a sampling spiking network (SSN) samples from its target distribution, we use the Kullback-Leibler divergence

$$D_\mathrm{KL}\left(p^\mathrm{net} \parallel p^\mathrm{target}\right) = \sum_{\boldsymbol{z}} p^\mathrm{net}_{\boldsymbol{z}} \ln \left(\frac{p^\mathrm{net}_{\boldsymbol{z}}}{p^\mathrm{target}_{\boldsymbol{z}}} \right) , \tag{7}$$

which is a measure for the similarity between the sampled distribution p^net and the target distribution p^target. For inference tasks, we determine the network's classification rate on a subset of the used data set which was put aside during training. Furthermore, generative properties of SSNs are investigated either by letting the network complete partially occluded examples from the data set or by letting it generate new examples.

Learning algorithm

Networks were trained with a Hebbian wake-sleep algorithm

$$\Delta \boldsymbol{W}_{ij} = \eta \left[p^\mathrm{target}_{z_i=1,z_j=1} - p^\mathrm{net}_{z_i=1,z_j=1} \right] , \tag{8}$$

$$\Delta \boldsymbol{b}_i = \eta \left[p^\mathrm{target}_{z_i=1} - p^\mathrm{net}_{z_i=1} \right] , \tag{9}$$

which minimizes the $D_\mathrm{KL}\left(p^\mathrm{net} \parallel p^\mathrm{target}\right)$ [50]. η is a learning rate which is either constant or decreases over time $\eta \propto \frac{1}{t}$. For high-dimensional data sets (e.g. handwritten letters and digits), binary-unit networks were trained with the CAST algorithm [51], a variant of wake-sleep with a tempering scheme, and then translated to SSN parameters with Eqs. (5) and (6) instead of training the SSNs directly to reduce simulation time.

Results

We approach the problem of externally-induced stochasticity incrementally. Throughout the remainder of the manuscript, we discern between background input, which is provided by other functional networks, and explicit noise, for which we use the conventional assumption of Poisson spike trains. We start by analyzing the effect of correlated background on the performance of SSNs. We then demonstrate how the effects of both auto- and cross-correlated background can be mitigated by Hebbian plasticity. This ultimately enables us to train a fully deterministic network of networks to perform different inference tasks without requiring any form of explicit noise.

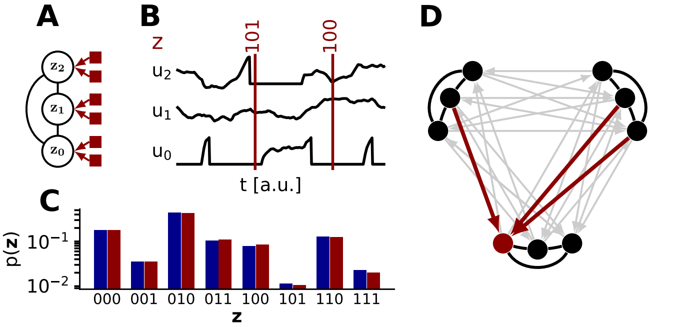

Figure 1: Sampling spiking networks (SSNs) with and without explicit noise. **(A)** Schematic of a sampling spiking network, where each neuron (circles) encodes a binary random variable $z_i \in \{0, 1\}$. In the original model, neurons were rendered effectively stochastic by adding external Poisson sources of high-frequency balanced noise (red boxes). **(B)** A neuron represents the state $z_k = 1$ when refractory and $z_k = 0$ otherwise. **(C)** The dynamics of neurons in an SSN can be described as sampling (red bars) from a target distribution (blue bars). **(D)** Instead of using Poisson processes as a source of explicit noise, we replace the Poisson input with spikes coming from other networks performing spike-based probablistic inference by creating a sparse, asymmetric connectivity matrix between several SSNs. For instance, the red neuron receives not only information-carrying spikes from its home network (black lines), but also spikes from the other two SSNs as background (red arrows), and in turn projects back towards these networks.

Background autocorrelations

Unlike ideal Poisson sources, single spiking neurons produce autocorrelated spike trains, with the shape of the autocorrelation function (ACF) depending on their firing rate $p(z = 1)$ and refractory time τ_ref. For higher output rates, spike trains become increasingly dominated by bursts, i.e., sequences of equidistant spikes with an interspike interval (ISI) of ISI $\approx \tau_\mathrm{ref}$. These fixed structures also remain in a population, since the population autocorrelation is equal to the averaged ACFs of the individual spike trains.

We investigated the effect of such autocorrelations on the output statistics of SSNs by replacing the Poisson input in the ideal model with spikes coming from other SSNs. As opposed to Poisson noise, the autocorrelation $\mathcal{C}(\rho_i, \rho_i, \Delta) = \frac{\langle \rho_i(t) \rho_i(t+\Delta) \rangle - \langle \rho_i \rangle^2}{\mathrm{Var}(\rho_i)}$ of the SSN-generated background (Fig. 2B) is non-singular and influences the free membrane potential (FMP) distribution (Fig. 2C) and thereby activation function (Fig. 2D) of individual sampling neurons. With increasing firing rates (controlled by the bias of the neurons in the background SSNs), the number of significant peaks in the ACF increases as well:

$$\mathcal{C}(\rho_i, \rho_i, n\tau_\mathrm{ref}) \approx \sum_{k=1}^{\infty} e^{k \ln \bar{p}} \delta\left([n - k]\tau_\mathrm{ref}\right) , \tag{10}$$

where $\bar{p}$ is the probability for a burst to start. This regularity in the background input manifests itself in a reduced width

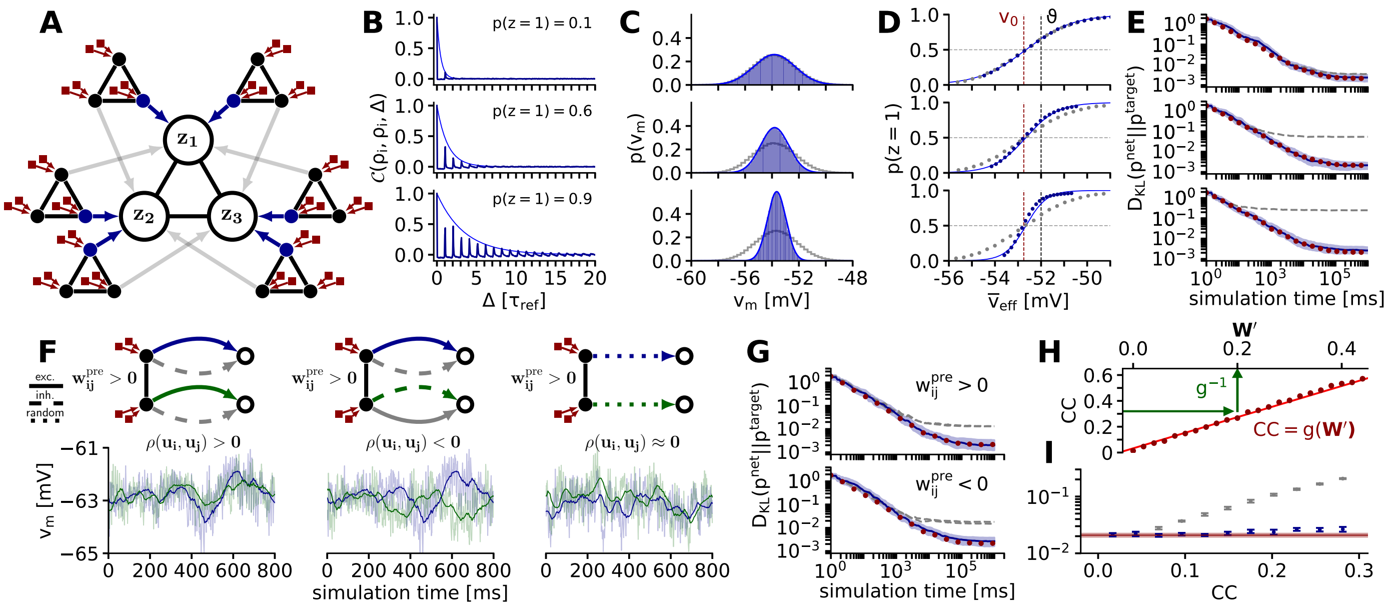

Figure 2: Effect of correlated background on SSN dynamics and compensation through reparametrization. **(A)** Feedforward replacement of Poisson noise by spiking activity from other SSNs. In this illustration, the principal SSN consists of three neurons receiving background input only from other functional SSNs that sample from their own predetermined target distribution. For clarity, only two out of a total of [260, 50, 34] (top to bottom in (B)) background SSNs per neuron are shown here. By modifying the background connectivity (gray and blue arrows) the amount of cross-correlation in the background input can be controlled. At this stage, the background SSNs are rendered stochastic by Poisson input (red boxes). **(B)** By appropriate parametrization of the background SSNs, we adjust the mean spike frequency of the background neurons (blue) to study the effect of background autocorrelations $\mathcal{C}(\rho_i, \rho_i, \Delta)$. Higher firing probabilities increase the chance of evoking bursts, which induce background autocorrelations for the neurons in the principal SSN at multiples of τ_{ref} (dark blue: simulation results; light blue: $e^{k \ln \bar{p}}$ with $k = \frac{\Delta}{\tau_{\text{ref}}}$, see Eq. (10)). **(C)** Background autocorrelation narrows the FMP distribution of neurons in the principal SSN: simulation (blue bars) and the theoretical prediction (Eq. (11), blue line) vs. background Poisson noise of the same rate (gray). Background intensities correspond to (B). **(D)** Single-neuron activation functions corresponding to (B,C) and the theoretical prediction (Eq. (12), blue line). For autocorrelated noise, the slope of the response curve changes, but the inflection point (with $p(z=1) = 0.5$) is conserved. **(E)** Kullback-Leibler divergence $D_{\text{KL}}\left(p^{\text{net}} \parallel p^{\text{target}}\right)$ (median and range between the first and third quartile) for the three cases shown in (B,C,D) after sampling from 50 different target distributions with 10 different random seeds for the 3-neuron network depicted in (A). Appropriate reparametrization can fully cancel out the effect of background autocorrelations (blue). The according results without reparametrization (gray) and with Poisson input (red) are also shown. **(F)** A pair of interconnected neurons in a background SSN generates correlated noise, as given by Eq. (13). The effect of cross-correlated background on a pair of target neurons depends on the nature of synaptic projections from the background to the principal SSN. Here, we depict the case where their interaction $\mathrm{w}_{\mathrm{ij}}^{\mathrm{pre}}$ is excitatory; the inhibitory case is a mirror image thereof. Left: If forward projections are of the same type, postsynaptic potentials will be positively correlated. Middle: Different synapse types in the forward projection only change the sign of the postsynaptic potential correlations. Right: For many background inputs with mixed connectivity patterns, correlations can average out to zero even when all input correlations have the same sign. **(G)** Same experiment as in (E), with background connection statistics adjusted to compensate for input cross-correlations. The uncompensated cases from (F, left) and (F, middle) are shown in gray. **(H)** Correlation-cancelling reparametrization in the principal SSN. By transforming the state space from $\boldsymbol{z} \in \{0,1\}^n$ to $\boldsymbol{z}' \in \{-1,1\}$, input correlations attain the same functional effect as synaptic weights (Eq. (15)); simulation results given as red dots, linear fit as red line. Weight rescaling followed by a transformation back into the $\boldsymbol{z} \in \{0,1\}^n$ state space, shown in green (which affects both weights and biases) can therefore alleviate the effects of correlated background. **(I)** Similar experiment as in (E) for a network with ten neurons, with parameters adjusted to compensate for input cross-correlations. As in the case of autocorrelated background, cross-correlations can be cancelled out by appropriate reparametrization.

σ' of the FMP distribution

$$f(u_i^{\text{free}}) \sim \mathcal{N}(\mu' = \mu, \sigma' = \sqrt{\beta}\sigma) \tag{11}$$

with a scaling factor $\sqrt{\beta}$ that depends on the ACF, which in turn translates to a steeper activation function

$$p(z_i = 1) \approx \int_{\vartheta}^{\infty} f(u)\mathrm{d}u \approx \varphi(u_0' = u_0, \alpha' = \sqrt{\beta}\alpha)\,, \tag{12}$$

with inflection point u_0' and inverse slope α'. Thus, autocorrelations in the background input lead to a reduced width of the FMP distribution and hence to a steeper activation function compared to the one obtained using uncorrelated Poisson input. For a better intuition, we used an approximation of the activation function of LIF neurons, but the argument also holds for the exact expression derived in [43], as verified by simulations (Fig. 2D).

Apart from the above effect, the background autocorrelations do not affect neuron properties that depend linearly on the synaptic noise input, such as the mean FMP and the inflection point of the activation function (equivalent to zero bias). Therefore, the effect of the background autocorrelations can be functionally reversed by rescaling the functional (from other neurons in the principal SSN) afferent synaptic weights by a factor equal to the ratio between the new and the original slope α'/α (Eqs. (5) and (6)), as shown in Fig. 2E.

Background cross-correlations

In addition to being autocorrelated, background input to pairs of neurons can be cross-correlated as well, due to either shared inputs or synaptic connections between the neurons that generate said background. These background cross-correlations can manifest themselves in a modified cross-correlation between the outputs of neurons, thereby distorting the distribution sampled by an SSN.

However, depending on the number and nature of presynaptic background sources, background cross-correlations may cancel out to a significant degree. The correlation coefficient (CC) of the FMPs of two neurons fed by correlated noise amounts to

$$\rho(u_i, u_j) \propto \sum_{l,m} w_l w_m \left(E_l^{\mathrm{rev}} - \mu_i\right)\left(E_m^{\mathrm{rev}} - \mu_j\right) \qquad (13)$$
$$\cdot \int \mathrm{d}\Delta \; \lambda_{li,mj} \; \mathcal{C}\left(S_{l,\mathrm{i}}, S_{m,\mathrm{j}}, \Delta\right) \widetilde{\mathcal{C}}\left(\kappa, \kappa, \Delta\right) \; ,$$

where l sums over all background spike trains $S_{l,i}$ projecting to neuron i and m sums over all background spike trains $S_{m,j}$ projecting to neuron j. $\widetilde{\mathcal{C}}\left(\kappa, \kappa, \Delta\right)$ is the unnormalized autocorrelation function of the postsynaptic potential (PSP) kernel κ, i.e., $\widetilde{\mathcal{C}}\left(\kappa, \kappa, \Delta\right) = \langle \kappa(t)\kappa(t+\Delta)\rangle$, and $\mathcal{C}\left(S_{l,\mathrm{i}}, S_{m,\mathrm{j}}, \Delta\right)$ the cross-correlation function of the background inputs. $\lambda_{li,mj}$ is given by $\lambda_{li,mj} = \sqrt{\mathrm{Var}\left(S_{l,\mathrm{i}}\right) \mathrm{Var}\left(S_{m,\mathrm{j}}\right)}$. The background cross-correlation is gated into the cross-correlation of FMPs by the nature of the respective synaptic connections: if the two neurons connect to the cross-correlated inputs by synapses of different type (one excitatory, one inhibitory), the sign of the CC is switched (Fig. 2F). However, individual contributions to the FMP CC also depend on the difference of the mean free membrane potential and the reversal potentials, so the gating of cross-correlations is not symmetric for excitatory and inhibitory synapses. Nevertheless, it is apparent that if the connectivity statistics (in-degree and synaptic weights) from the background sources to an SSN are chosen appropriately and enough presynaptic partners are available, the total pairwise cross-correlation between neurons in an SSN can cancel out to zero, leaving the sampling performance unimpaired (Fig. 2G). It is important to note that this way of reducing cross-correlations is independent of the underlying weight distribution of the networks providing the background; the required cross-wiring of functional networks could therefore, in principle, be encoded genetically and does not need to be learned. Furthermore, a very simple cross-wiring rule, i.e., independently and randomly determined connections, already suffices to accomplish low background cross-correlations and therefore reach a good sampling performance.

While this method is guaranteed to work in an artificial setting, further analysis is needed to assess its compatibility with the cortical connectome with respect to connectivity statistics or synaptic weight distributions. However, even if cortical architecture prevents a clean implementation of this decorrelation mechanism, SSNs can themselves compensate for residual background cross-correlations by modifying their parameters, similar to the autocorrelation compensation discussed above.

To demonstrate this ability, we need to switch from the natural state space of neurons $\boldsymbol{z} \in \{0,1\}^N$ to the more symmetric space $\boldsymbol{z}' \in \{-1,1\}^N$.[1] By requiring $p(\boldsymbol{z}') \overset{!}{=} p(\boldsymbol{z})$ to conserve state probabilities (and thereby also correlations), the desired change of state variables $\boldsymbol{z}' = 2\boldsymbol{z} - \mathbf{1}$ can be achieved with a linear parameter transformation:

$$\boldsymbol{W}' = \frac{1}{4}\boldsymbol{W} \text{ and } \boldsymbol{b}' = \frac{1}{2}\boldsymbol{b} + \frac{1}{4}\sum_i \mathrm{col}_i \boldsymbol{W} \, . \qquad (14)$$

In the $\{-1,1\}^N$ state space, both synaptic connections w'_{ij} and background cross-correlations $\rho(s_i, s_j)$ shift probability mass from the mixed states $(z_i, z_j) = (-1, 1)$ and $(1, -1)$ to the aligned states $(z_i, z_j) = (-1, -1)$ and $(1, 1)$ (see SI, Fig. S1). Therefore, by adjusting $\boldsymbol{b}$ and $\boldsymbol{W}$, it is possible to find a $\boldsymbol{W}'$ (Fig. 2H) that precisely conserves the desired correlation structure between neurons:

$$w'_{ij} = g^{-1}[\rho(s_i, s_j)] \approx \frac{\rho(s_i, s_j) - g_0}{g_1} \, , \qquad (15)$$

with constants g_0 and g_1 (Fig. 2I). Therefore, when an SSN learns a target distribution from data, background cross-correlations are equivalent to an offset in the initial network parameters and are automatically compensated during training.

At this point, we can conclude that all effects that follow from replacing input noise in an SSN with functional output from other SSNs (which still receive explicit noise) can be compensated by appropriate parameter adjustments. This is an important preliminary conclusion for the next sections, where we show how all noise can be eliminated in an ensemble of interconnected SSNs endowed with synaptic plasticity without significant penalty to their respective functional performance. We start with larger ensembles of small networks, each of which receives its own target distribution, which allows a straightforward quantitative assessment of their sampling performance $D_{\mathrm{KL}}\left(p^{\mathrm{net}} \parallel p^{\mathrm{target}}\right)$. We study the behavior of such ensembles both in computer simulations and on mixed-signal neuromorphic hardware. Finally, we demonstrate the capability of our approach for truly functional, larger-scale networks, trained on high-dimensional visual data.

Sampling without explicit noise in large ensembles

We initialized an ensemble of 100 6-neuron SSNs with an inter-network connectivity of $\epsilon = 0.1$ and random synaptic weights. No external input is needed to kick-start network activity, as some neurons spike spontaneously, due to the random initialization of parameters (see Fig. 3A). The existence of inhibitory weights disrupts the initial regularity, initiating

[1] The $z = 0$ state for a silent neuron is arguably more natural, because it has no effect on its postsynaptic partners during this state. In contrast, $z \in \{-1, 1\}$ would, for example, imply efferent excitation upon spiking and constant efferent inhibition otherwise.

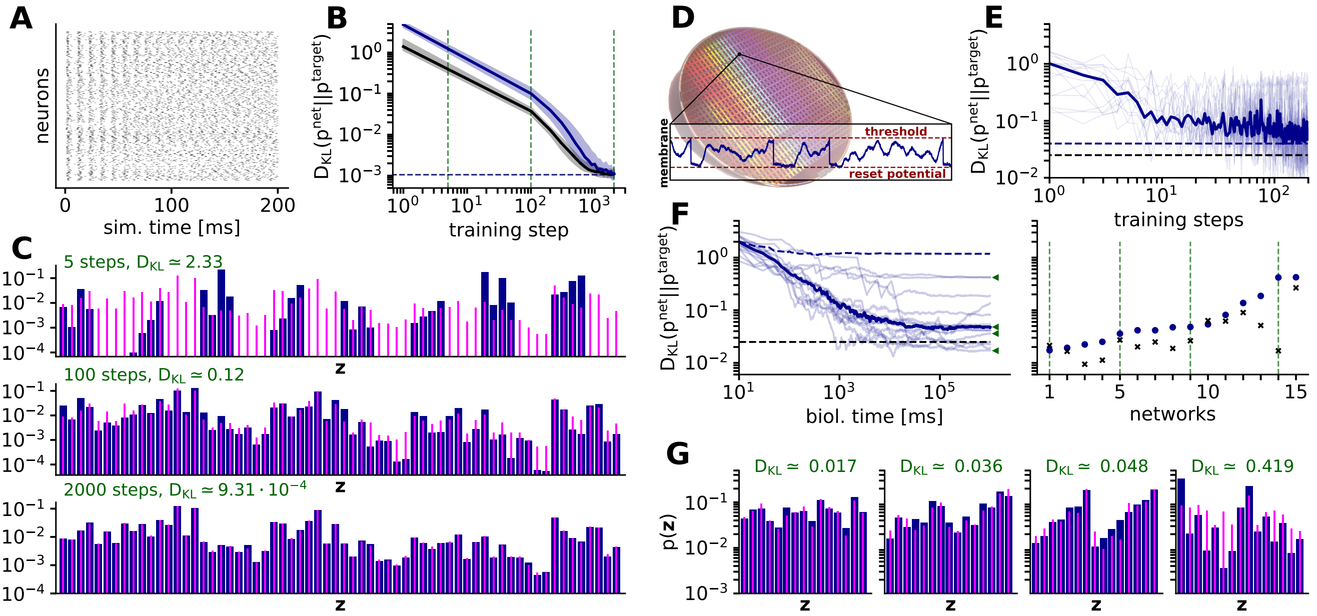

Figure 3: Sampling without explicit noise from a set of predefined target distributions in software (A-C) and on a neuromorphic substrate (D-G). **(A)** Temporal evolution of spiking activity in an ensemble of 100 interconnected 6-neuron SSNs with no source of explicit noise. An initial burst of regular activity caused by neurons with a strong enough positive bias quickly transitions to asynchronous irregular activity due to inhibitory synapses. **(B)** Median sampling quality of the above ensemble during learning. At the end of the learning phase, the sampling quality of individual networks in the ensemble (blue) is on par with the one obtained in the theoretically ideal case of independent networks with Poisson background (black). Error bars given over 5 simulation runs with different random seeds. **(C)** Illustration of a single target distribution (magenta) and corresponding sampled distribution (blue) of a network in the ensemble at several stages of the learning process. **(D)** Photograph of a wafer from the BrainScaleS neuromorphic system used in (E), (F) and (G) before post-processing (i.e., adding additional structures like buses on top), which would mask the underlying modular structure. Blue: exemplary membrane trace of an analog neuron receiving Poisson noise. **(E)** Performance of an ensemble consisting of 15 4-neuron SSNs with no external noise during learning on the neuromorphic substrate, shown in light blue for each SSN and with the median shown in dark blue. The large fluctuations compared to (B) are a signature of the natural variability of the substrate's analog components. The dashed blue line represents the best achieved median performance at $D_{\mathrm{KL}}\left(p^{\mathrm{net}} \parallel p^{\mathrm{target}}\right) = 3.99 \times 10^{-2}$. For comparison, we also plot the optimal median performance for the theoretically ideal case of independent, Poisson-driven SSNs emulated on the same substrate, which lies at $D_{\mathrm{KL}}\left(p^{\mathrm{net}} \parallel p^{\mathrm{target}}\right) = 2.49 \times 10^{-2}$ (dashed black line). **(F)** Left: Demonstration of sampling in the neuromorphic ensemble of SSNs after 200 training steps. Individual networks in light blue, median performance in dark blue. Dashed blue line: median performance before training. Dashed black line: median performance of ideal networks, as in (E). Right: Best achieved performance, after 100 s of bio time (10 ms of hardware time) for all SSNs in the ensemble depicted as blue dots (sorted from lowest to highest D_{KL}). For comparison, the same is plotted as black crosses for their ideal counterparts. **(G)** Sampled (blue) and target (magenta) distributions of four of the 15 SSNs. The selection is marked in (F) with green triangles (left) and vertical green dotted lines (right). Since we made no particular selection of hardware neurons according to their behavior, hardware defects have a significant impact on a small subset of the SSNs. Despite these imperfections, a majority of SSNs perform close to the best value permitted by the limited weight resolution (4 bits) of the substrate.

the sampling process.

Ongoing learning (Equations (8) and (9)) shapes the sampled distributions towards their respective targets (Fig. 3B), the parameters of which were drawn randomly. Our ensemble achieved a sampling performance (median D_{KL}) of $1.06^{+0.27}_{-0.40} \times 10^{-3}$, which is similar to the median performance of an idealized setup (independent, Poisson-driven SSNs) of $1.05^{+0.15}_{-0.35} \times 10^{-3}$ (errors are given by the first and third quartile). To put the above D_{KL} values in perspective, we compare the sampled and target distributions of one of the SSNs in the ensemble at various stages of learning (Fig. 3C). Instead of training ensembles, they can also be set up by translating the parameters of the target distributions to neurosynaptic parameters directly, as discussed in the previous section (see SI, Fig. S2).

As we show in the following, this approach to noise-free sampling-based computation can also be applied to physical neural substrates which incorporate unreliable components and are therefore significantly more difficult to control.

Implementation on a neuromorphic substrate

To test the robustness of our results, we studied an implementation of noise-free sampling on an artificial neural substrate. For this, we used the BrainScaleS system [52], a mixed-signal neuromorphic platform with analog neurosynaptic dynamics and digital inter-neuron communication (Fig. 3D, see also SI, Fig. S3). A major advantage of this implementation is the emulation speedup of 10^4 with respect to biological real-time; however, for clarity, we shall continue using biological time units instead of actual emulation time.

The additional challenge for our neuronal ensemble is to cope with the natural variability of the substrate, caused mainly by fixed-pattern noise, or with other limitations such as a finite weight resolution (4 bits) or spike loss, which can all be substantial [53, 54]. It is important to note that the ability to function when embedded in an imperfect substrate with significant deviations from an idealized model represents a necessary prerequisite for viable theories of biological neural function.

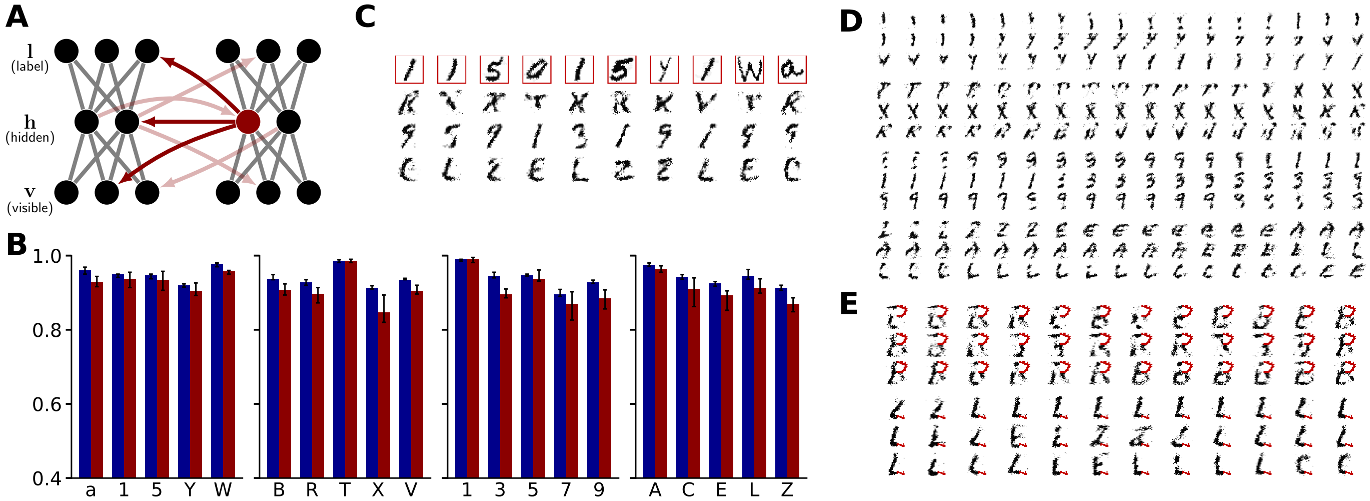

Figure 4: Bayesian inference on visual input. **(A)** Illustration of the connectivity between two hierarchical SSNs in the simulated ensemble. Each SSN had a visible layer **v**, a hidden **h** and a label layer **l**. Neurons in the same layer of an SSN were not interconnected. Each neuron in an SSN received only activity from the hidden layers of other SSNs as background (no sources of explicit noise). **(B)** An ensemble of four such SSNs (red) was trained to perform generative and discriminative tasks on visual data from the EMNIST dataset. We used the classification rate of restricted Boltzmann machines trained with the same hyperparameters as a benchmark (blue). Error bars are given (on blue) over 10 test runs and (on red) over 10 ensemble realizations with different random seeds. **(C)** Illustration of a scenario where one of the four SSNs (red boxes) received visual input for classification (B). At the same time, the other SSNs continuously generated images from their respective learned distributions. **(D)** Pattern generation and mixing during unconstrained dreaming. Here, we show the activity of the visible layer of all four networks from (B), each spanning three rows. Time evolves from left to right. For further illustrations of the sampling process in the ensemble of hierarchical SSNs, see SI, Fig. S4, S5. **(E)** Pattern completion and rivalry for two instances of incomplete visual stimulus. The stimulus consisted of the top right and bottom right quadrant of the visible layer, respectively. In the first run, we clamped the top arc of a "B" compatible with either a "B" or an "R" (top three rows, red), in the second run we chose the bottom line of an "L" compatible with an "L", an "E", a "Z" or a "C" (bottom three rows, red). An ensemble of SSNs performs Bayesian inference by implicitly evaluating the conditional distribution of the unstimulated visible neurons, which manifests itself here as sampling from all image classes compatible with the ambiguous simulus (see also SI, Fig. S6).

We emulated an ensemble of 15 4-neuron SSNs, with an inter-SSN connectivity of $\epsilon = 0.2$ and with randomly drawn target distributions. The biases were provided by additional bias neurons and adjusted during learning via the synaptic weights between bias and sampling neurons, along with the synapses within the SSNs, using the same learning rule as before (Equations (8) and (9)). After 200 training steps, the ensemble reached a median D_{KL} of $3.99^{+1.27}_{-1.15} \cdot 10^{-2}$ (errors given by the distance to the first and third quartile) compared to $1.18^{+0.47}_{-0.55}$ before training (Fig. 3E). As a point of reference, we also considered the idealized case by training the same set of SSNs without interconnections and with every neuron receiving external Poisson noise generated from the host computer, reaching a D_{KL} of $2.49^{+3.18}_{-0.71} \cdot 10^{-2}$.

This relatively small performance loss of the noise-free ensemble compared to the ideal case confirms the theoretical predictions and simulation results. Importantly, this was achieved with only a rather small ensemble, demonstrating that large numbers of neurons are not needed for realizing this computational paradigm.

In Fig. 3F, we show the sampling dynamics of all emulated SSNs after learning. While most SSNs are able to approximate their target distributions well, some sampled distributions are significantly skewed (Fig. 3G). This is caused by a small subset of dysfunctional neurons, which we have not discarded beforehand, in order to avoid an implausibly fine-tuned use-case of the neuromorphic substrate. These effects become less significant in larger networks trained on data instead of predefined distributions, where learning can naturally cope with such outliers by assigning them smaller output weights. Nevertheless, these results demonstrate the feasibility of self-sustained Bayesian computation through sampling in physical neural substrates, without the need for any source of explicit noise. Importantly, and in contrast to other approaches [55], every neuron in the ensemble plays a functional role, with no neuronal real-estate being dedicated to the production of (pseudo-)randomness.

Ensembles of hierarchical SSNs

When endowed with appropriate learning rules, hierarchical spiking networks can be efficiently trained on high-dimensional visual data [47, 54, 56–59]. Such hierarchical networks are characterized by the presence of several layers, with connections between consecutive layers, but no lateral connections within the layers themselves. When both feedforward and feedback connections are present, such networks are able to both classify and generate images that are similar to those used during training.

In these networks, information processing in both directions is Bayesian in nature. Bottom-up propagation of information enables an estimation of the conditional probability of a particular label to fit the input data. Additionally, top-down propagation of neural activity allows generating a subset of patterns in the visible layer conditioned on incomplete or partially occluded visual stimulus. When no input is presented, such networks will produce patterns similar to those enforced during training ("dreaming"). In general, the exploration of a multimodal solution space in generative models is facilitated by some noise-generating mechanism. We demonstrate how even a small interconnected set of hierarchical SSNs can perform these computations self-sufficiently, without any source of explicit noise.

We used an ensemble of four 3-layer hierarchical SSNs trained on a subset of the EMNIST dataset [60], an extended version of the widely used MNIST dataset [61] that includes digits as well as capital and lower-case letters. All SSNs had the same structure, with 784 visible units, 200 hidden units and 5 label units (Fig. 4A). To emulate the presence of networks with different functionality, we trained each of them on a separate subset of the data. (To combine sampling in space with sampling in time, multiple networks can also be trained on the same data, see SI Fig. S5.) Since training the spiking ensemble directly was computationally prohibitive, we trained four Boltzmann machines on the respective datasets and then translated the resulting parameters to neurosynaptic parameters of the ensemble using the analytical approximations for correlation compensation described earlier in the manuscript.

To test the discriminative properties of the SSNs in the ensemble, one was stimulated with visual input, while the remaining three were left to freely sample from their underlying distribution. We measured a median classification rate of $91.5^{+3.6}_{-3.0}\%$ with errors given by the distance to the first and third quartile, which is close to the $94.0^{+2.1}_{-1.5}\%$ achieved by the idealized reference setup provided by the abstract Boltzmann machines (Fig. 4B). At the same time, all other SSNs remained capable of generating recognizable images (Fig. 4C). It is expected that direct training and a larger number of SSNs in the ensemble would further improve the results, but a functioning translation from the abstract to the biological domain already underpins the soundness of the underlying theory.

Without visual stimulus, all SSNs sampled freely, generating images similar to those on which they were trained (Fig. 4D). Without any source of explicit noise, the SSNs were capable to mix between the relevant modes (images belonging to all classes) of their respective underlying distributions, which is a hallmark of a good generative model. We further extended these results to an ensemble trained on the full MNIST dataset, reaching a similar generative performance for all networks (see SI Fig. S5).

To test the pattern completion capabilities of the SSNs in the ensemble, we stimulated them with incomplete and ambiguous visual data (Fig. 4E). Under these conditions, SSNs only produced images compatible with the stimulus, alternating between different image classes, in a display of pattern rivalry. As in the case of free dreaming, the key mechanism facilitating this form of exploration was provided by the functional activity of other neurons in the ensemble.

Discussion

Based on our findings, we argue that sampling-based Bayesian computation can be implemented in ensembles of spiking networks without requiring any explicit noise-generating mechanism. While in biology various explicity sources of noise exist [62–64], these forms of stochasticity are either too weak (in case of ion channels) or too high-dimensional for efficient exploration (in the case of stochastic synaptic transmission, as used for, e.g., reinforcement learning [65]). On the other hand, neuronal population noise can be highly correlated, affecting information processing by, e.g., inducing systematic sampling biases [32].

In our proposed framework, each network in an ensemble plays a dual role: while fulfilling its assigned function within its home subnetwork, it also provides its peers with the spiking background necessary for stochastic search within their respective solution spaces. This enables a self-consistent and parsimonious implementation of neural sampling, by allowing all neurons to take on a functional role and not dedicating any resources purely to the production of background stochasticity. The underlying idea lies in adapting neuro-synaptic parameters by (contrastive) Hebbian learning to compensate for auto- and cross-correlations induced by interactions between the functional networks in the ensemble. Importantly, we show that this does not rely on the presence of a large number of independent presynaptic partners for each neuron, as often assumed by models of cortical computation that use Poisson noise (see, e.g., [66]). Instead, only a small number of ensembles is necessary to implement noise-free Bayesian sampling. This becomes particularly relevant for the development of neuromorphic platforms by eliminating the computational footprint imposed by the generation and distribution of explicit noise, thereby reducing power consumption and bandwidth constraints.

The suggested noise-free Bayesian brain reconciles the debate on spatial versus temporal sampling [67, 41]. In fact, the suggested ensembles of spiking neurons that provide each other with virtual noise may be arranged in parallel sensory streams. An ambiguous stimulus will trigger different representations on each level of these streams, forming a hierarchy of probabilistic population codes. While these population codes learn to cover the full sensory distribution in space, they will also generate samples of the sensory distribution in time (see Fig. S5 in the SI). Attention may select the most likely representation, while suppressing the representations in the other streams. Analogously, possible actions may be represented in parallel motor streams during planning and a motor decision may select the one to be performed. When recording in premotor cortex, such a selection causes a noise reduction [68], that we suggest is effectively the signature of choosing the most probable action in a Bayesian sense.

In our simulations, we have used a simplified neuron model to reduce computation time and facilitate the mathematical analysis. However, we expect the core underlying principles to generalize, as evidenced by our results on neuromorphic hardware, where the dynamics of individual neurons and synapses differ significantly from the mathematical model. Such an ability to compute with unreliable components represents a particularly appealing feature in the context of both biology and emerging nanoscale technologies.

Acknowledgments

We thank Luziwei Leng, Nico Gürtler and Johannes Bill for valuable discussions. We further thank Eric Müller and Christian Mauch for maintenance of the computing cluster we used for simulations and Luziwei Leng for providing code implementing the CAST algorithm. This work has received funding from the European Union 7th Framework Programme under grant agreement 604102 (HBP), the Horizon 2020 Framework Programme under grant agreement 720270 (HBP) and the Manfred Stärk Foundation.

References

[1] GH Henry, PO Bishop, RM Tupper, and B Dreher. Orientation specificity and response variability of cells in the striate cortex. *Vision research*, 13(9):1771–1779, 1973.

[2] Peter H Schiller, Barbara L Finlay, and Susan F Volman. Short-term response variability of monkey striate neurons. *Brain research*, 105(2):347–349, 1976.

[3] Rufin Vogels, Werner Spileers, and Guy A Orban. The response variability of striate cortical neurons in the behaving monkey. *Experimental brain research*, 77(2):432–436, 1989.

[4] Robert J Snowden, Stefan Treue, and Richard A Andersen. The response of neurons in areas v1 and mt of the alert rhesus monkey to moving random dot patterns. *Experimental Brain Research*, 88(2):389–400, 1992.

[5] Amos Arieli, Alexander Sterkin, Amiram Grinvald, and AD Aertsen. Dynamics of ongoing activity: explanation of the large variability in evoked cortical responses. *Science*, 273(5283):1868, 1996.

[6] Rony Azouz and Charles M Gray. Cellular mechanisms contributing to response variability of cortical neurons in vivo. *Journal of Neuroscience*, 19(6):2209–2223, 1999.

[7] Rajesh PN Rao, Bruno A Olshausen, and Michael S Lewicki. *Probabilistic models of the brain: Perception and neural function*. MIT press, 2002.

[8] Konrad P Körding and Daniel M Wolpert. Bayesian integration in sensorimotor learning. *Nature*, 427(6971):244–247, 2004.

[9] Jan W Brascamp, Raymond Van Ee, Andre J Noest, Richard HAH Jacobs, and Albert V van den Berg. The time course of binocular rivalry reveals a fundamental role of noise. *Journal of vision*, 6(11):8–8, 2006.

[10] Gustavo Deco, Edmund T Rolls, and Ranulfo Romo. Stochastic dynamics as a principle of brain function. *Progress in neurobiology*, 88(1):1–16, 2009.

[11] József Fiser, Pietro Berkes, Gergő Orbán, and Máté Lengyel. Statistically optimal perception and learning: from behavior to neural representations. *Trends in cognitive sciences*, 14(3):119–130, 2010.

[12] Wolfgang Maass. Searching for principles of brain computation. *Current Opinion in Behavioral Sciences*, 11:81–92, 2016.

[13] Richard B Stein. Some models of neuronal variability. *Biophysical journal*, 7(1):37–68, 1967.

[14] Nicolas Brunel. Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journal of computational neuroscience*, 8(3):183–208, 2000.

[15] Nicolas Fourcaud and Nicolas Brunel. Dynamics of the firing probability of noisy integrate-and-fire neurons. *Neural computation*, 14(9):2057–2110, 2002.

[16] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[17] DK Smetters and Anthony Zador. Synaptic transmission: noisy synapses and noisy neurons. *Current Biology*, 6(10):1217–1218, 1996.

[18] Wolfgang Maass and Anthony M Zador. Dynamic stochastic synapses as computational units. In *Advances in neural information processing systems*, pages 194–200, 1998.

[19] Yosef Yarom and Jorn Hounsgaard. Voltage fluctuations in neurons: signal or noise? *Physiological reviews*, 91(3):917–929, 2011.

[20] Rubén Moreno-Bote. Poisson-like spiking in circuits with probabilistic synapses. *PLoS computational biology*, 10(7):e1003522, 2014.

[21] Emre O Neftci, Bruno U Pedroni, Siddharth Joshi, Maruan Al-Shedivat, and Gert Cauwenberghs. Stochastic synapses enable efficient brain-inspired learning machines. *Frontiers in neuroscience*, 10:241, 2016.

[22] Charles F Stevens and Anthony M Zador. When is an integrate-and-fire neuron like a poisson neuron? In *Advances in neural information processing systems*, pages 103–109, 1996.

[23] Hans E Plesser and Wulfram Gerstner. Noise in integrate-and-fire neurons: from stochastic input to escape rates. *Neural computation*, 12(2):367–384, 2000.

[24] EJ Chichilnisky. A simple white noise analysis of neuronal light responses. *Network: Computation in Neural Systems*, 12(2):199–213, 2001.

[25] Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[26] Peter Dayan, LF Abbott, et al. Theoretical neuroscience: computational and mathematical modeling of neural systems. *Journal of Cognitive Neuroscience*, 15(1):154–155, 2003.

[27] Srdjan Ostojic and Nicolas Brunel. From spiking neuron models to linear-nonlinear models. *PLoS computational biology*, 7(1):e1001056, 2011.

[28] Elad Schneidman, Barry Freedman, and Idan Segev. Ion channel stochasticity may be critical in determining the reliability and precision of spike timing. *Neural computation*, 10(7):1679–1703, 1998.

[29] Peter N Steinmetz, Amit Manwani, Christof Koch, Michael London, and Idan Segev. Subthreshold voltage noise due to channel fluctuations in active neuronal membranes. *Journal of computational neuroscience*, 9(2):133–148, 2000.

[30] Moritz Deger, Moritz Helias, Clemens Boucsein, and Stefan Rotter. Statistical properties of superimposed stationary spike trains. *Journal of Computational Neuroscience*, 32(3):443–463, 2012.

[31] JI Nelson, PA Salin, MH-J Munk, M Arzi, and J Bullier. Spatial and temporal coherence in cortico-cortical connections: a cross-correlation study in areas 17 and 18 in the cat. *Visual neuroscience*, 9(1):21–37, 1992.

[32] Bruno B Averbeck, Peter E Latham, and Alexandre Pouget. Neural correlations, population coding and computation. *Nature reviews neuroscience*, 7(5):358, 2006.

[33] Emilio Salinas and Terrence J Sejnowski. Correlated neuronal activity and the flow of neural information. *Nature reviews neuroscience*, 2(8):539, 2001.

[34] Ronen Segev, Morris Benveniste, Eyal Hulata, Netta Cohen, Alexander Palevski, Eli Kapon, Yoash Shapira, and Eshel Ben-Jacob. Long term behavior of lithographically prepared in vitro neuronal networks. *Physical review letters*, 88(11):118102, 2002.

[35] József Fiser, Chiayu Chiu, and Michael Weliky. Small modulation of ongoing cortical dynamics by sensory input during natural vision. *Nature*, 431(7008):573, 2004.

[36] Robert Rosenbaum, Tatjana Tchumatchenko, and Rubén Moreno-Bote. Correlated neuronal activity and its relationship to coding, dynamics and network architecture. *Frontiers in computational neuroscience*, 8:102, 2014.

[37] Rubén Moreno-Bote, Alfonso Renart, and Néstor Parga. Theory of input spike auto-and cross-correlations and their effect on the response of spiking neurons. *Neural computation*, 20(7):1651–1705, 2008.

[38] Zachary F Mainen and Terrence J Sejnowski. Reliability of spike timing in neocortical neurons. *Science*, 268(5216):1503, 1995.

[39] Anthony Zador. Impact of synaptic unreliability on the information transmitted by spiking neurons. *Journal of Neurophysiology*, 79(3):1219–1229, 1998.

[40] Alexander Rauch, Giancarlo La Camera, Hans-Rudolf Lüscher, Walter Senn, and Stefano Fusi. Neocortical pyramidal cells respond as integrate-and-fire neurons to in vivo–like input currents. *Journal of neurophysiology*, 90(3):1598–1612, 2003.

[41] Gergő Orbán, Pietro Berkes, József Fiser, and Máté Lengyel. Neural variability and sampling-based probabilistic representations in the visual cortex. *Neuron*, 92(2):530–543, 2016.

[42] Lars Buesing, Johannes Bill, Bernhard Nessler, and Wolfgang Maass. Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons. *PLoS Comput Biol*, 7(11):e1002211, 2011.

[43] Mihai A. Petrovici, Johannes Bill, Ilja Bytschok, Johannes Schemmel, and Karlheinz Meier. Stochastic inference with spiking neurons in the high-conductance state. *Phys. Rev. E*, 94:042312, Oct 2016.

[44] Dejan Pecevski, Lars Buesing, and Wolfgang Maass. Probabilistic inference in general graphical models through sampling in stochastic networks of spiking neurons. *PLoS computational biology*, 7(12):e1002294, 2011.

[45] Dimitri Probst, Mihai A Petrovici, Ilja Bytschok, Johannes Bill, Dejan Pecevski, Johannes Schemmel, and Karlheinz Meier. Probabilistic inference in discrete spaces can be implemented into networks of lif neurons. *Frontiers in computational neuroscience*, 9, 2015.

[46] Galit Fuhrmann, Idan Segev, Henry Markram, and Misha Tsodyks. Coding of temporal information by activity-dependent synapses. *Journal of neurophysiology*, 87(1):140–148, 2002.

[47] Luziwei Leng, Roman Martel, Oliver Breitwieser, Ilja Bytschok, Walter Senn, Johannes Schemmel, Karlheinz Meier, and Mihai A Petrovici. Spiking neurons with short-term synaptic plasticity form superior generative networks. *Scientific Reports*, 8(1):10651, 2018.

[48] Andrew Davison, Daniel BrÃČÂijderle, Jochen Eppler, Jens Kremkow, Eilif Muller, Dejan Pecevski, Laurent Perrinet, and Pierre Yger. Pynn: a common interface for neuronal network simulators. *Frontiers in Neuroinformatics*, 2:11, 2009.

[49] Marc-Oliver Gewaltig and Markus Diesmann. Nest (neural simulation tool). *Scholarpedia*, 2(4):1430, 2007.

[50] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

[51] Ruslan Salakhutdinov. Learning deep boltzmann machines using adaptive mcmc. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 943–950, 2010.

[52] Johannes Schemmel, Johannes Fieres, and Karlheinz Meier. Wafer-scale integration of analog neural networks. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 431–438. IEEE, 2008.

[53] Mihai A Petrovici, Bernhard Vogginger, Paul Müller, Oliver Breitwieser, Mikael Lundqvist, Lyle Muller, Matthias Ehrlich, Alain Destexhe, Anders Lansner, René Schüffny, et al. Characterization and compensation of network-level anomalies in mixed-signal neuromorphic modeling platforms. *PloS one*, 9(10):e108590, 2014.

[54] Sebastian Schmitt, Johann Klähn, Guillaume Bellec, Andreas Grübl, Maurice Guettler, Andreas Hartel, Stephan Hartmann, Dan Husmann, Kai Husmann, Sebastian Jeltsch, et al. Neuromorphic hardware in the loop: Training a deep spiking network on the brainscales wafer-scale system. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 2227–2234. IEEE, 2017.

[55] Jakob Jordan, Mihai A Petrovici, Oliver Breitwieser, Johannes Schemmel, Karlheinz Meier, Markus Diesmann, and Tom Tetzlaff. Stochastic neural computation without noise. *arXiv preprint arXiv:1710.04931*, 2017.

[56] Mihai A Petrovici, Sebastian Schmitt, Johann Klähn, David Stöckel, Anna Schroeder, Guillaume Bellec, Johannes Bill, Oliver Breitwieser, Ilja Bytschok, Andreas Grübl, et al. Pattern representation and recognition with accelerated analog neuromorphic systems. In *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*, pages 1–4. IEEE, 2017.

[57] Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training deep spiking neural networks using backpropagation. *Frontiers in neuroscience*, 10:508, 2016.

[58] Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation*, 30(6):1514–1541, 2018.

[59] Saeed Reza Kheradpisheh, Mohammad Ganjtabesh, Simon J Thorpe, and Timothée Masquelier. Stdp-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, 99:56–67, 2018.

[60] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.

[61] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[62] A Aldo Faisal, Luc PJ Selen, and Daniel M Wolpert. Noise in the nervous system. *Nature reviews neuroscience*, 9(4):292, 2008.

[63] Tiago Branco and Kevin Staras. The probability of neurotransmitter release: variability and feedback control at single synapses. *Nature Reviews Neuroscience*, 10(5):373, 2009.

[64] John A White, Jay T Rubinstein, and Alan R Kay. Channel noise in neurons. *Trends in neurosciences*, 23(3):131–137, 2000.

[65] H Sebastian Seung. Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron*, 40(6):1063–1073, 2003.

[66] Xiaohui Xie and H Sebastian Seung. Learning in neural networks by reinforcement of irregular spiking. *Physical Review E*, 69(4):041909, 2004.

[67] Wei Ji Ma, Jeffrey M Beck, Peter E Latham, and Alexandre Pouget. Bayesian inference with probabilistic population codes. *Nature neuroscience*, 9(11):1432, 2006.

[68] Mark M Churchland, M Yu Byron, Stephen I Ryu, Gopal Santhanam, and Krishna V Shenoy. Neural variability in premotor cortex provides a signature of motor preparation. *Journal of Neuroscience*, 26(14):3697–3712, 2006.

[69] Ilja Bytschok, Dominik Dold, Johannes Schemmel, Karlheinz Meier, and Mihai A. Petrovici. Spike-based probabilistic inference with correlated noise. *BMC Neuroscience*, 2017.

[70] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Supporting Information

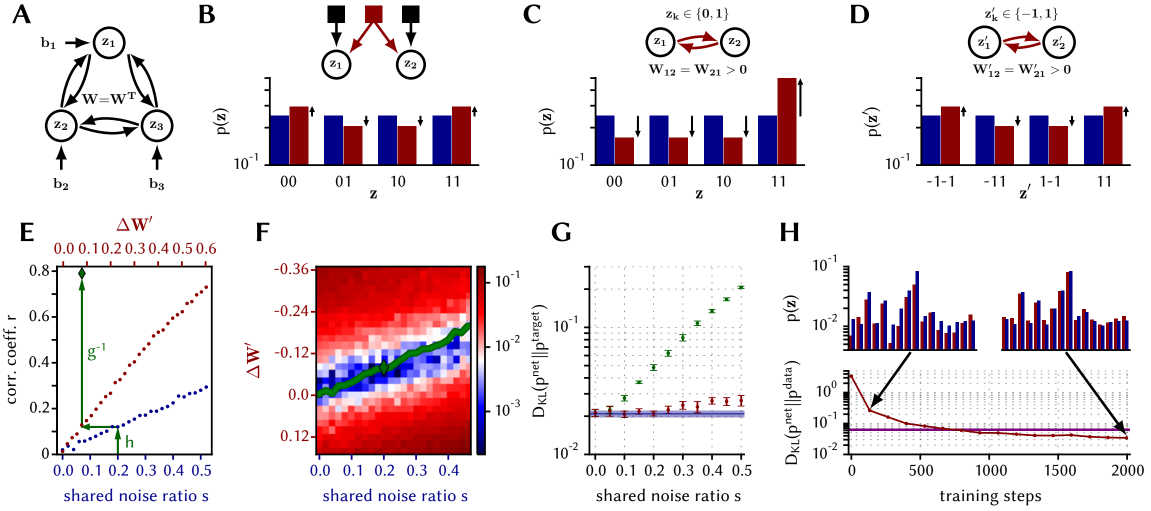

Figure S1: Compensation of input correlations by adjustment of weights and biases in an SSN. For simplicity, this is illustrated here for the case of shared input correlations, but the results hold for all types of statically correlated inputs. See also [69] for additional information. **(A)** Exemplary architecture of a network with 3 neurons that samples from a Boltzmann distribution with parameters $\mathbf{W}$ and $\mathbf{b}$. In order to achieve the required stochastic regime, each neuron receives external noise in the form of Poisson spike trains (not shown). **(B)-(D)** Exemplary sampled distributions for a network of two neurons. The "default" case is the one where all weights and biases are set to zero (uniform distribution, blue bars). **(B)** Shared noise sources have a correlating effect, shifting probability mass into the (1,1) and (0,0) states (red bars). **(C)** In the $\{0,1\}^2$ space, increased weights introduce a (positive) shift of probability mass from all other states towards the (1,1) state (red bars), which is markedly different from the effect of correlated noise. **(D)** In the $\{-1,1\}^2$ space, increased weights have the same effect as correlated noise (red bars). **(E)** Dependence of the correlation coefficient r between the states of two neurons on the change in synaptic weight $\Delta W'$ (red) and the shared noise ratio s (blue). These define bijective functions g and h that can be used to compute the weight change ($\Delta W' = f(s), \quad$ with $f := g^{-1} \circ h$) needed to compensate the effect of correlated noise in the $\{-1,1\}^N$ space. **(F)** Study of the optimal compensation rule in a network with two neurons. For simplicity, the ordinate represents weight changes for a network with states in the $\{-1,1\}^2$ space, which are then translated to corresponding parameters $(\mathbf{W}, \mathbf{b})$ for the $\{0,1\}^2$ state space. The colormap shows the difference between the sampled and the target distribution measured by the Kullback-Leibler divergence $D_{\mathrm{KL}}\left(p^{\mathrm{net}} \parallel p^{\mathrm{target}}\right)$. The mapping provided by the compensation rule f (see (E)) is depicted by the green curve. Note that the compensation rule $\Delta W' = f(s)$ provides a nearly optimal parameter translation. Remaining deviations are due to differences between LIF and Glauber dynamics. **(G)** Compensation of noise correlations in an SSN with ten neurons. The results are depicted for a set of ten randomly drawn Boltzmann distributions over $\boldsymbol{z} \in \{0,1\}^{10}$ (error bars). For a set of randomly chosen Boltzmann distributions, a ten-neuron network performs sampling in the presence of pairwise-shared noise ratios s (x-axis). The blue line marks the sampling performance without noise-induced correlations ($s = 0$). For an increasing shared noise ratio, uncompensated noise (green) induces a significant increase in sampling error. After compensation, the sampling performance is nearly completely restored. As before, remaining deviations are due to differences between LIF and Glauber dynamics. **(H)** An LIF-based ten-neuron network with shared noise sources ($s = 0.3$ for each neuron pair) is trained with data samples generated from a target Boltzmann distribution (blue bars). During training, the sampled distribution becomes an increasingly better approximation of the target distribution (red line). For comparison, we also show the distribution sampled by an SSN with parameters translated directly from the Boltzmann parameters (purple). The trained network is able to improve upon this result because learning implicitly compensates for the abovementioned differences between LIF and Glauber dynamics.

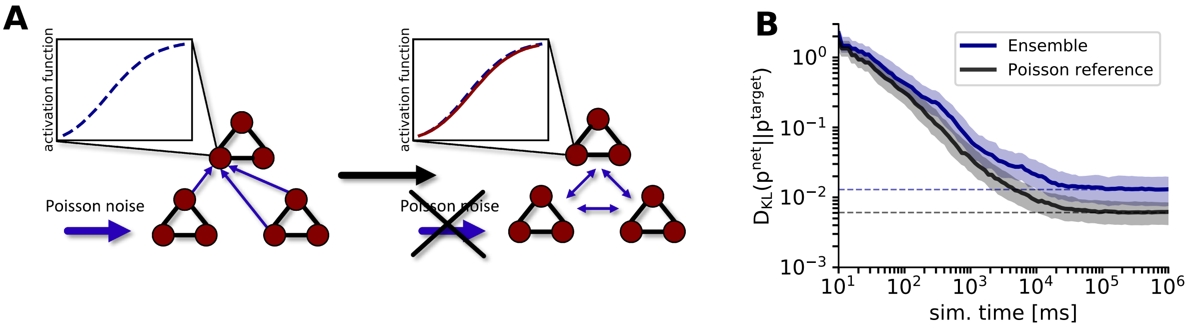

Figure S2: **(A)** A straightforward way to set up the parameters of each network (w_{ij} and E_l) is to use the parameter translation as described in the main text, i.e., use the corresponding activation function of each neuron to correctly account for the background noise statistics. This is demonstrated here for the case of (left) 399 networks (only two shown) receiving Poisson noise and one network only receiving ensemble input and (right) all networks only receiving ensemble input. In both cases, the resulting activation function is the same and we can indeed use it to translate the parameters of the target distribution to neurosynaptic parameters. **(B)** Using the corresponding activation functions to set up the ensemble (but no training), each network in the ensemble is indeed able to accurately sample from its target distribution without explicit noise, as expected from our considerations in (A) and the main text. This is shown here (in software simulations) for an ensemble of 400 3-neuron SSNs with an interconnection probability of 0.05, reaching a median D_{KL} of $12.8^{+6.4}_{-5.0} \times 10^{-3}$ (blue), which is close to the ideal result with Poisson noise of $6.2^{+2.0}_{-2.0} \times 10^{-3}$ (black, errors given as the first and third quartile).

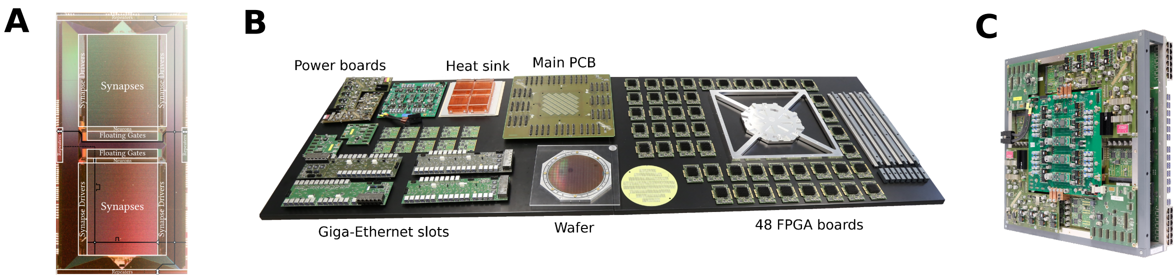

Figure S3: **(A)** A single HICANN chip (High Input Count Analog Neural Network), the elemental building block of the BrainScaleS wafer. The HICANN consists of two symmetric halves and harbors analog implementations of adaptive exponential integrate-and-fire (AdEx) neurons and conductance-based synapses in 180nm CMOS technology. Floating gates next to the neuron circuits are used to store neuron parameters. Spikes are routed digitally through horizontal and vertical buses (not shown) and translated into postsynaptic conductances in the synapse array. Unlike in simulations on general-purpose CPUs, here neurons and synapses are physically implemented, with no numeric computations being performed to calculate network dynamics. A single wafer consists of 384 HICANN chips. **(B)** Individual components of the BrainScaleS system, including both wafer and support structure. For instance, FPGA boards provide an I/O interface for wafer configuration and spike data and Giga-Ethernet slots provide a connection between FPGAs and the control cluster from which users conduct their experiments via Python scripts using the PyNN API. **(C)** Completely assembled wafer of the BrainScaleS neuromorphic system.

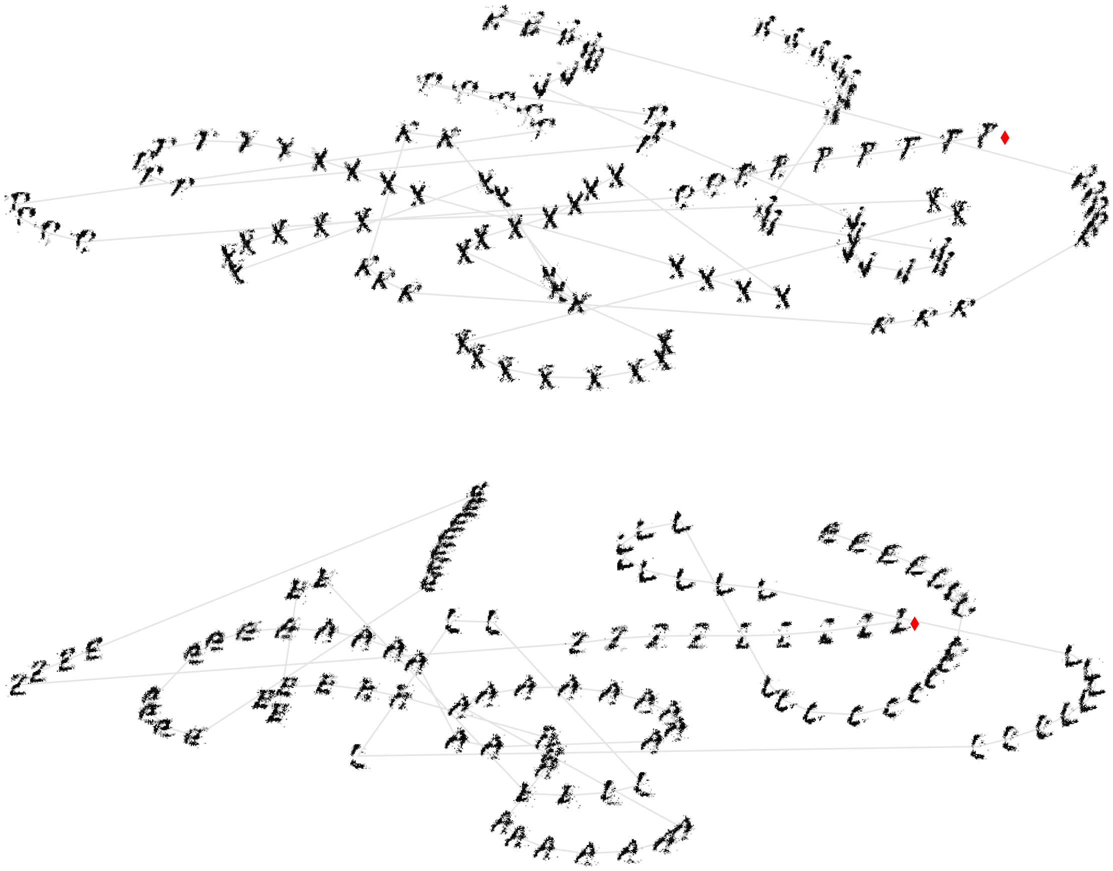

Figure S4: t-SNE representation [70] of consecutively generated images of two of the four SSNs trained on EMNIST. Both SSNs smoothly traverse several regions of the state space representing image classes while dreaming. The red diamond marks the first image in the sequence, gray lines connect consecutive images. Consecutive images are 200 ms apart.

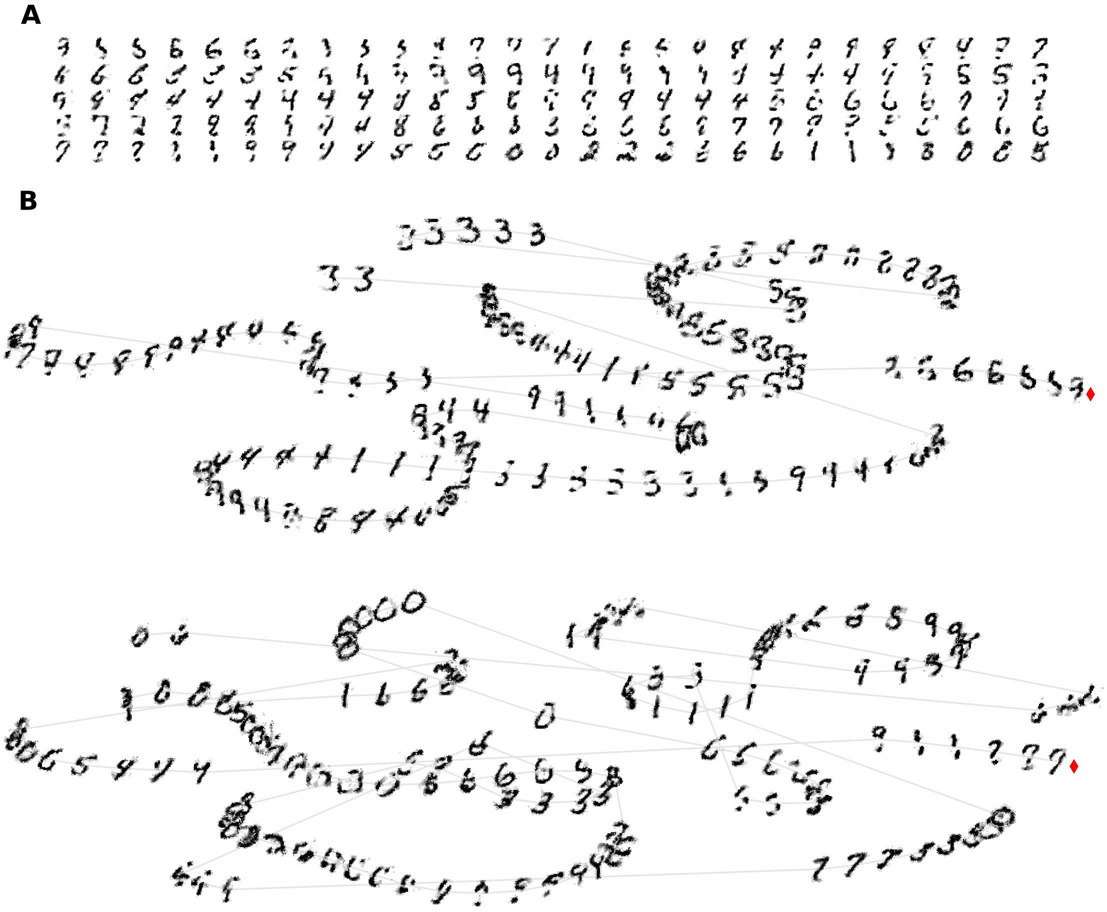

Figure S5: **(A)** Dreaming ensemble of five hierarchical SSNs with 784 visible, 500 hidden and 10 label neurons (without explicit noise). Each row represents samples from a single network of the ensembles, with samples being $375\,\mathrm{ms}$ apart. To set up the ensemble, a restricted Boltzmann machine was trained on the MNIST dataset and the resulting parameters translated to corresponding neurosynaptic parameters of the ensemble. Here, to facilitate mixing, we used short-term depression to modulate synaptic interactions and weaken attractor states that would be otherwise difficult to escape [47]. **(B)** t-SNE representation [70] of consecutively generated images of two of the five SSNs trained on MNIST digits. Both SSNs are able to generate and mix between diverse images of different digit classes while dreaming. The red diamond marks the first image in the sequence, gray lines connect consecutive images. Consecutive images are $400\,\mathrm{ms}$ apart.

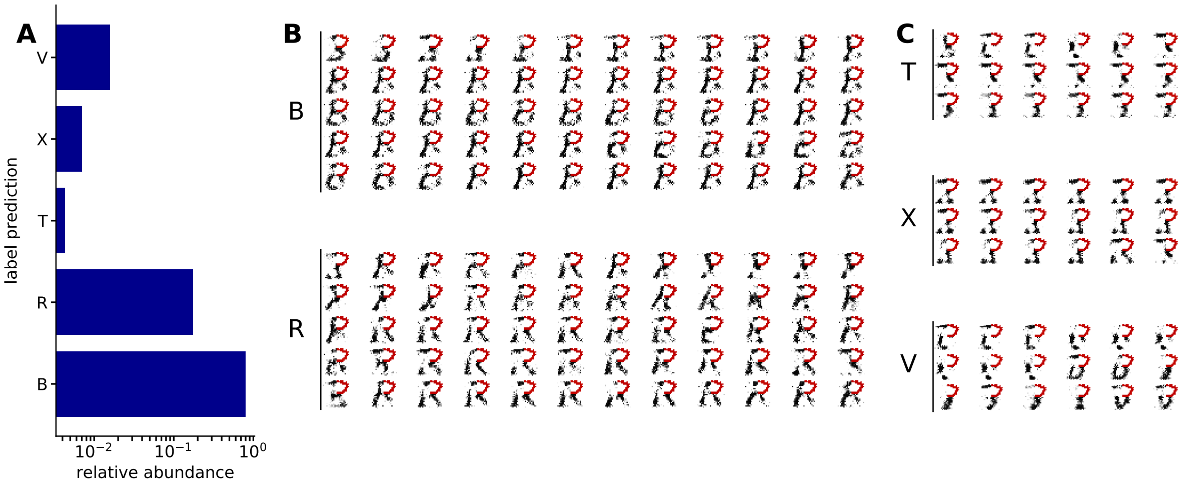

Figure S6: **(A)** Relative abundance of the label output while clamping parts of a "B". Most of the time (**79.85%**), the image is correctly classified as a "B". The closest alternative explanation, an "R", is generated second most (**17.45%**). The remaining classes are explored significantly less often by the network (**0.43%, 0.70%, 1.57%**). **(B)** Examples of the visible layer activity while the label layer classifies the partially clamped images either as a "B" (top) or an "R" (bottom). **(C)** Examples of the visible layer activity while classifying the image as a "T", "X" or "V". In these cases, the images generated by the visible neurons show prominent features of these letters.