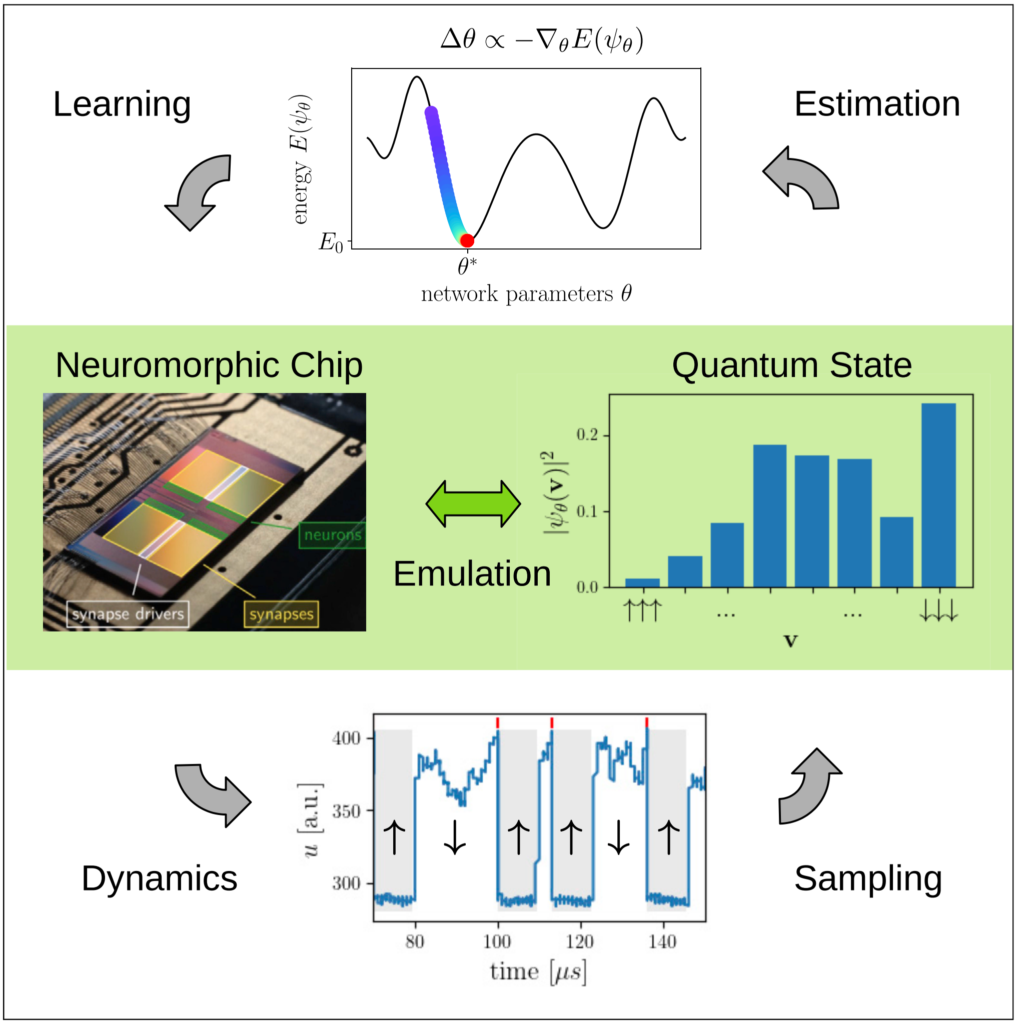
iScience

CellPress
OPEN ACCESS

Article

Variational learning of quantum ground states on spiking neuromorphic hardware



Robert Klassert, Andreas Baumbach, Mihai A. Petrovici, Martin Gärttner

andreas.baumbach@kip.uni-heidelberg.de

Highlights

Variational scheme for representing quantum ground states with neuromorphic hardware

Accelerated physical system yields system-size independent sample generation time

Accurate learning of ground states across a quantum phase transition

Detailed analysis of algorithmic and technical limitations

Klassert et al., iScience *25*, 104707
August 19, 2022 © 2022 The Author(s).
https://doi.org/10.1016/j.isci.2022.104707

iScience

CellPress OPEN ACCESS

Article

Variational learning of quantum ground states on spiking neuromorphic hardware

Robert Klassert,[1] Andreas Baumbach,[1,2,*] Mihai A. Petrovici,[2,1] and Martin Gärttner[3,1,4]

SUMMARY

Recent research has demonstrated the usefulness of neural networks as variational ansatz functions for quantum many-body states. However, high-dimensional sampling spaces and transient autocorrelations confront these approaches with a challenging computational bottleneck. Compared to conventional neural networks, physical model devices offer a fast, efficient and inherently parallel substrate capable of related forms of Markov chain Monte Carlo sampling. Here, we demonstrate the ability of a neuromorphic chip to represent the ground states of quantum spin models by variational energy minimization. We develop a training algorithm and apply it to the transverse field Ising model, showing good performance at moderate system sizes ($N \leq 10$). A systematic hyperparameter study shows that performance depends on sample quality, which is limited by temporal parameter variations on the analog neuromorphic chip. Our work thus provides an important step towards harnessing the capabilities of neuromorphic hardware for tackling the curse of dimensionality in quantum many-body problems.

INTRODUCTION

The Hilbert space of quantum many-body systems and consequently the computational resources required to describe them grow exponentially with system size. On the one hand, this poses a challenge to understanding collective quantum effects, for example in condensed matter physics (Avella and Mancini, 2012; Zhou et al., 2021). On the other hand, efficient numerical tools are required for the characterization and validation of quantum devices such as digital quantum computers currently under development (Preskill, 2018). Fortunately, many physical systems exhibit symmetries and structure that allow to reduce the exponential complexity and to design tractable approaches for the representation of the wave function. For example, so-called stoquastic Hamiltonians are known to have positive ground state wave functions allowing the application of quantum Monte Carlo methods (Becca and Sorella, 2017). Locally interacting systems featuring an excitation gap have limited ground state entanglement, which renders tensor network states an efficient method for approximating them (Schollwöck, 2011). Such physical structure may, however, not always be easy to discover and exploit. Because the process of automatically discovering structure despite the curse of dimensionality is a discipline of machine learning, variational approaches using artificial neural networks (ANNs) have found their way into quantum many-body physics in recent years (Carrasquilla, 2020). These so-called neural quantum states (NQS) have been shown to serve as efficient function approximators that rival competing approaches like tensor networks by providing accurate representations for a large class of quantum states using only a small number of parameters. Among other applications NQS has been successfully employed as a variational ansatz for ground state search (Carleo and Troyer, 2017; Jia et al., 2019; Carrasquilla, 2020), quantum dynamics (Carleo and Troyer, 2017; Czischek et al., 2018; Hartmann and Carleo, 2019; Nagy and Savona, 2019; Schmitt and Heyl, 2020; Reh et al., 2021), and quantum state tomography (Torlai et al., 2018; Carrasquilla et al., 2019; Torlai and Melko, 2020).

The most successful existing variational approaches for representing many-body ground states rely on the use of Markov chain Monte Carlo (MCMC) methods to generate samples based on which observables are estimated (Melko et al., 2019). Probabilistic inference with MCMC in high-dimensional spaces comes with a number of associated challenges such as trading off accuracy against sample correlations and capturing multi-modality within short simulation times. In particular, the sampling of neural network quantum states

[1] Kirchhoff-Institut für Physik, Ruprecht-Karls-Universität Heidelberg, Im Neuenheimer Feld 227, 69120 Heidelberg, Germany

[2] Department of Physiology, University of Bern, 3012 Bern, Switzerland

[3] Physikalisches Institut, Ruprecht-Karls-Universität Heidelberg, Im Neuenheimer Feld 226, 69120 Heidelberg, Germany

[4] Institut für Theoretische Physik, Ruprecht-Karls-Universität Heidelberg, Philosophenweg 16, 69120 Heidelberg, Germany

*Correspondence: andreas.baumbach@kip.uni-heidelberg.de

https://doi.org/10.1016/j.isci.2022.104707

is known to be a computationally challenging task in the case of restricted Boltzmann machines (RBM) (Long and Servedio, 2010). To tackle this challenge we use a physical neurmorphic device which enables the fast generation of independent samples to approximate quantum wave functions.

We develop and demonstrate a method for approximating the ground states of quantum spin systems by variationally adapting the physical parameters of a neuromorphic hardware chip. The neuromorphic chip functions as a spiking neural network (SNN) emulator. Such networks work in a similar way to neuronal networks in the brain. We use the refractory state of a neuron (refractory, $z = 1$, or non-refractory, $z = 0$) to encode the state (up, ↑, or down, ↓) of a quantum spin. SNNs, in contrast to ANNs, have inherent time dynamics and process their inputs in an event-based fashion. Because of the physical implementation the emulation becomes inherently parallel, rendering the sampling speed independent of the network size. We note that neuromorphic hardware has recently been used to represent entangled quantum states using a mapping of general mixed quantum states to a probabilistic representation and training the system to represent a given state by approximating its corresponding probability distribution (Czischek et al., 2022). Here, instead, we directly encode the wave function of pure quantum states and use this approach for variational ground state search through minimization of the quantum system's total energy. Our state representation assumes positive real wave function coefficients, a property which is guaranteed for ground states of stoquastic Hamiltonians (Bravyi et al., 2008). Using the transverse field Ising model (TFIM) as a benchmark case, we find that its ground state can be represented accurately for any value of the transverse field including the quantum phase transition point.

We further study current limitations of our proposed approach. In particular, we investigate several technical limitations of our neuromorphic back-end in detail and pinpoint the main loci of potential improvement for future revisions. In addition, we discuss the algorithmic advantages and drawbacks of our sample-based representation.

Note that, unlike other functional tasks that SNNs have been employed for in the past (Petrovici et al., 2016; Kungl et al., 2019; Dold et al., 2019), which only require the reproduction of large scale features, for example, image classes, we require the full probability distribution to be sampled with high precision. We therefore demonstrate a new level of sampling precision for neuromorphic systems, which potentially opens up new applications beyond the specific one considered here. Our work serves as a demonstration of variational ground state learning on neuromorphic devices. This opens the door to adaptions using alternative, analog or digital neuromorphic hardware (Davies et al., 2018; Thakur et al., 2018; Roy et al., 2019), and the development of improved learning algorithms exploiting fast neuromorphic sample generation.

The remainder of this work is structured as follows: We begin by laying the foundations of spike-based computing (Section spike-based sampling) and the BrainScaleS-2 neuromorphic substrate (Section neuromorphic chip), followed by details about the variational algorithm, quantum state representation (Section variational algorithm) and the physical system, namely the TFIM (Section transverse-field Ising model), which it is applied to. In Section performance, we examine and discuss the performance of our approach and specifically investigate the dependence on system size. Section limitations provides a detailed analysis of the impact of hardware constraints on the performance of our method. We conclude in Section discussion and describe future research directions.

THEORETICAL AND EXPERIMENTAL METHODOLOGY

Spike-based sampling

Generative models based on ANN can be used to encode and sample from probability distributions (Ackley et al., 1985; Hinton et al., 1995). Similarly, SNNs can be shown to approximately implement Markov-chain Monte Carlo sampling, albeit with dynamics that differ fundamentally from standard statistical methods (Petrovici, 2016). Here, we use the BrainScaleS-2 neuromorphic platform (Billaudelle et al., 2020) to encode the wavefunction of quantum spin systems using the activity distribution of a two-layer network architecture (Figure 1). The implementation is inspired by Boltzmann machines (BM) in that the n network neurons encode binary values. The visible units $\boldsymbol{v} = (v_1, \ldots, v_N)$ are used to directly represent the quantum spin system and the hidden units $\boldsymbol{h} = (h_1, \ldots, h_{N_h})$ mediate correlations between spins. The full network state is the concatenation of visible and hidden units $\boldsymbol{z} = (\boldsymbol{v}, \boldsymbol{h})$.

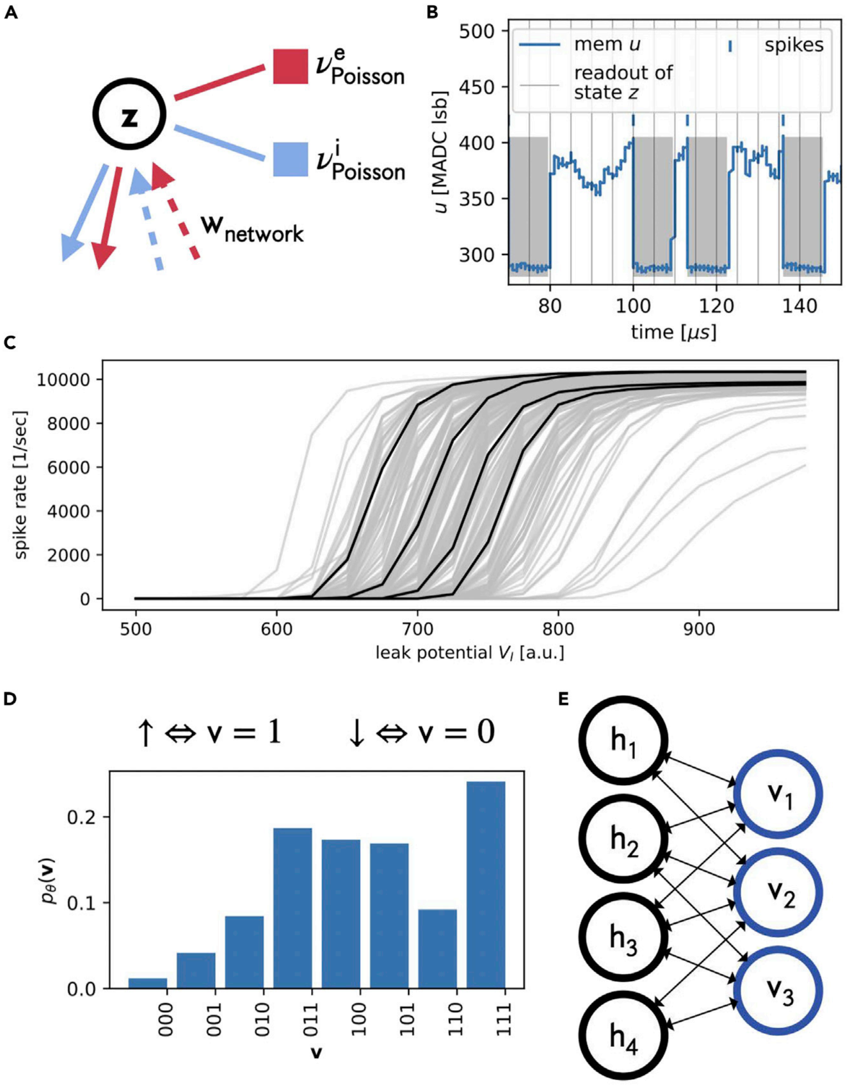


Figure 1. SNN sampling of quantum states

(A) A LIF neuron under Poisson stimulus forms a spiking sampling unit. For technical reasons excitatory (red) and inhibitory (blue) connections are implemented separately.

(B) Exemplary membrane potential evolution of a spiking sampling unit. Binary states are assigned according to the refractory state (shaded area $z = 1$, $z = 0$ otherwise), which overrides the membrane dynamics after emitting a spike (blue dashes). States are readout periodically (gray lines).

(C) Neuronal response functions of of all 192 neurons used. For better visibility four of these are plotted in black. Note that this diversity is because of the variability of the analog substrate; for a more in-depth discussion, we refer to Pfeil et al. (2013); Petrovici et al. (2014); Schmitt et al. (2017).

(D) Frequency of occurrence of neuron states retrieved as described in panel (b) approximating the model distribution $p_\theta(\mathbf{v})$. The visible states $\mathbf{v} \in \{0,1\}^N$ are identified with basis states $|\mathbf{v}\rangle \in \{|\downarrow\rangle, |\uparrow\rangle\}^{\otimes N}$ of the corresponding quantum spin system.

(E) Layered network architecture used throughout this manuscript.

We use leaky integrate-and-fire (LIF) neurons to implement our network. The dynamics of such neurons are governed by

$$C_\mathrm{m} \frac{du}{dt} = g_\mathrm{l}(V_\mathrm{l} - u) + I_\mathrm{syn}(t) \, , \qquad \text{(Equation 1)}$$

where C_m is the capacitance of the neuron's membrane, u its potential, V_l the leak potential it decays toward via the leak conductance g_l, and I_{syn} the total input current to the neuron. The input current is given by a weighted sum over spike-triggered interaction kernels $\kappa(t)$ for all spikes from all connected neurons. For a detailed discussion we refer to Methods D: Leaky integrate-and-fire neurons and Gerstner et al., 2002.

Whenever the membrane potential u of a neuron exceeds a threshold value V_{thresh}, it generates a spike and the membrane is short-circuited to a reset value $V_{reset} < V_{thresh}$ (Figure 1B). This reset implements the fixed refractory time τ_{ref} during which we consider the neuron to be in state $z = 1$ (gray shaded region in Figure 1B, state $z = 0$ otherwise). The generated spikes are routed to other neurons via synapses with interaction strength w.

Networks of such LIF neurons under Poisson stimulus (Figure 1A) can be shown to approximately sample from characteristic Boltzmann distributions (Petrovici et al., 2016). In this scenario, biological neurons enter a high-conductance state with a short membrane time constant $\tau_m = C_m/g_l \ll \tau_{ref}$, and their spike response function (Figure 1C) is well described by a logistic function

$$p(z = 1|V_l) = \frac{1}{1 + \exp(-[V_l - u_0]/\alpha)}, \qquad \text{(Equation 2)}$$

where u_0 represents the position of and α the slope at the inflection point – for a detailed derivation see Petrovici, 2016. Note, that changing V_l has the same effect as a change of the synaptic input I_{syn}. In other words, each neuron effectively calculates $p(z = 1|I_{syn})$, such that the network as a whole can be shown to approximately sample from a Boltzmann distribution $p_\theta(\boldsymbol{z}) = \exp[-\varepsilon_\theta(\boldsymbol{z})]$ with network energy $\varepsilon_\theta(\boldsymbol{z}) = -\sum_{i,j} z_i W_{ij} z_j/2 - \sum_i z_i b_i$ and parameters $\theta = (\boldsymbol{W}, \boldsymbol{b})$.

One can relate the abstract weights W_{ij} to the physical strength of the synaptic interaction w_{ij} from neuron i to neuron j and the abstract biases b_i to the value of the physical leak potential V_l of each neuron. These two parameter domains are related linearly but have different units. The mapping between physical neuron and synapse parameters and abstract Boltzmann weights can be gauged by measuring the logistic activation function (Equation 2) with respect to some form of current stimulus. This relation neglects some dynamic aspects and as such only holds approximately (Petrovici, 2016). This does not restrict the learning scheme applied here. The probability distribution of physical interest is then the marginal over the hidden space (Figures 1D and 1E)

$$p_\theta(\boldsymbol{v}) = \frac{1}{Z_\theta} \sum_{\boldsymbol{h}} \exp[-\varepsilon_\theta(\boldsymbol{z})], \qquad \text{(Equation 3)}$$

which is used to encode the ground state wave function (see Section variational algorithm). The partition sum $Z_\theta = \sum_{\boldsymbol{z}} p_\theta(\boldsymbol{z})$ ensures proper normalization.

Neuromorphic chip

We used the BrainScaleS-2-HICANN-X-v2 physical neuromorphic system (Billaudelle et al., 2020) – in the following abbreviated as BSS-2 – depicted in Figure 2A, for all experiments reported in this manuscript. BSS-2 is a mixed-signal neuromorphic chip, with 512 adaptive exponential leaky integrate-and-fire (AdEx) neuron circuits, which we configured to implement current-based LIF neurons (see Equation 1). Because of their analog nature, neuron dynamics are 1000 times faster than in their biological counterparts. Spikes are communicated as digital events which then trigger an analog post-synaptic interaction in downstream neurons. For more details see Methods A: Description of the BrainScaleS-2 ASIC or Pehle et al., 2022.

We employed a routing protocol that forms a freely configurable network of 256 spike sources, combining two neuronal circuits in order to increase their maximum number of presynaptic sources to 256. We assigned 64 of these to the on-chip (noise) spike generators to provide a pool of stochasticity required for sampling (Petrovici, 2016). The full on-chip network structure, including both the sampling network and the noise source allocation, is shown in Figure 2B. The bipartite connection graph is reflected in the block structure of the connection matrix (left part) and the noise sources are randomly assigned from a fixed pool of 32 excitatory and 32 inhibitory sources (right part). This left us with up to 192 arbitrarily connectable stochastic neurons of which we used a subset to variationally learn the probability distribution representing the ground state wave function of a physical system of interest (see Figures 1D and 1E).

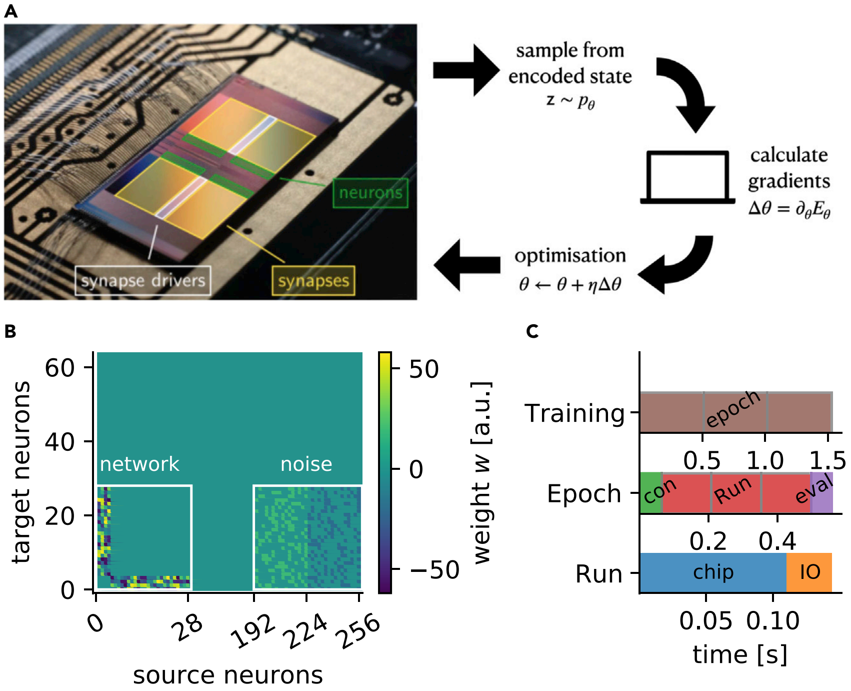


Figure 2. Neuromorphic learning scheme

(A) BrainScaleS-2 neuromorphic chip. It emulates the accelerated dynamics of up to 512 spiking LIF neurons. The learning algorithm alternates between on-chip neural sampling and off-chip gradient calculation that informs the network parameter updates $\Delta\theta$ to minimize the energy of the represented state.

(B) Exemplary synaptic weight matrix $\mathbf{w}$ for $N = 8$ and $N_h = 20$. Unused network parts (inputs 28 to 192 and neurons 64 to 192) are omitted for better visibility. The layered network structure manifests itself in the block structure of the lower left connectivity matrix. Each neuron is randomly assigned 10 out of the 64 possible noise sources (right part).

(C) Distribution of the wall clock time spent during an experimental run. Each epoch (brown) starts with a (partial) reconfiguration of the chip (con, green), followed by a number of consecutive sampling runs (red), followed by the evaluation (eval, purple) which includes the gradient calculation (see Table in STAR Methods). Each hardware run consists of actual chip execution (chip, blue) and a transfer to the host (IO, orange).

For each hardware run the BSS-2 chip returns a list of all (output) spike times and associated neuron IDs. This information combined with the measured τ_{ref} for each neuron is sufficient to reconstruct the network state $\boldsymbol{z}(t)$ at any point in time t. We computed the network state at regular intervals, as visualized in Figure 1B. The resulting binary configurations were collected in a histogram as shown in Figure 1D and formed an estimate of the steady-state distribution $p(\boldsymbol{z})$ of the current network configuration. By identifying the neuronal states ($z \in \{0, 1\}$) with the basis states of a qubit system ($|\uparrow\rangle, |\downarrow\rangle$) (see Figure 1D), $p(\boldsymbol{z})$ represents the quantum many-body state. Treating the physical network parameters as variational parameters this representation can be tuned to the ground state of a quantum system, as detailed in the following.

Variational algorithm

Our goal is to find an approximation of the ground state of a given stoquastic Hamiltonian H. For this we need to determine the parameter set θ for which our variational anzatz $|\psi_\theta\rangle$ of the ground state wave function minimizes the expectation value of the energy:

$$E_\theta = \langle\psi_\theta|H|\psi_\theta\rangle. \qquad \text{(Equation 4)}$$

The restriction to stoquastic Hamiltonians guarantees that the wave function of the corresponding ground state has non-negative real coefficients in the chosen basis which is the case if all off-diagonal elements of the Hamiltonian are negative (Bravyi et al., 2008). We use this property to directly relate the probability distribution $p_\theta(\boldsymbol{v})$ to the wave function coefficients, such that

$$|\psi_\theta\rangle = \sum_{\mathbf{v}} \sqrt{p_\theta(\mathbf{v})}|\mathbf{v}\rangle, \qquad \text{(Equation 5)}$$

where $p_\theta(\mathbf{v})$ is estimated by the relative frequency of the occurrence of $\mathbf{v}$ in the samples generated by the SNN (see Equation 3 and Figures 1D and 1E) as discussed above.

We employ a gradient-based minimization of the variational energy E_θ. Differentiating Equation (4) with respect to the parameters $\theta = (W_{ij}, b_i)$ results in (see Methods B: Derivation of the learning rule for details)

$$\partial_{W_{ij}} E_\theta = \left\langle \left(E_{\mathbf{v}}^{\text{loc}} - E_\theta \right) z_i z_j \right\rangle_{p_\theta(\mathbf{z})} \qquad \text{(Equation 6)}$$

$$\partial_{b_k} E_\theta = \left\langle \left(E_{\mathbf{v}}^{\text{loc}} - E_\theta \right) z_k \right\rangle_{p_\theta(\mathbf{z})}, \qquad \text{(Equation 7)}$$

where

$$E_{\mathbf{v}}^{\text{loc}} = \sum_{\mathbf{v}'} H_{\mathbf{v}\mathbf{v}'} \sqrt{p_\theta(\mathbf{v}')} \Big/ \sqrt{p_\theta(\mathbf{v})} \qquad \text{(Equation 8)}$$

is the local energy. Evaluating the local energies requires access to an estimate of the probabilities $p_\theta(\mathbf{v}')$ for all states $\mathbf{v}'$ for which the matrix element $H_{\mathbf{v}\mathbf{v}'} = \langle \mathbf{v}|H|\mathbf{v}'\rangle$ of the Hamiltonian is non-vanishing. Because no analytical relation between the physical parameters of the spiking network and the abstract parameters θ of the assumed RBM distribution is known, we estimate the probabilities $p_\theta(\mathbf{v})$ from samples. In particular, this means that we need to iterate through the whole collection of generated samples $\{\mathbf{z}\} \sim p_\theta$ twice. Once to generate the estimate for $p_\theta(\mathbf{v}')$ and once to calculate the averages in Equation (6) and Equation (7).

We implement a gradient descent scheme by alternating between the neuromorphic sampling $\{\mathbf{z}\} \sim p_\theta$ and host-based gradient calculations (see Figure 2A). In each iteration the chip is reconfigured according to the gradient given in Equation (6) and Equation (7) using the ADAM optimizer (Kingma and Ba, 2014), see Methods C: Adaptive momentum optimization for details. Each training iteration consists of a single hardware (re)configuration followed by multiple sampling runs of 0.1 s each, which corresponds to $2 \cdot 10^5$ independent samples, and subsequent gradient calculation (see Figure 2C for relative timings). We emphasize that only the evaluation part scales with the size of the used network and thereby the represented system, whereas the sampling time itself is system-size-independent.

In order to track the accuracy of the algorithm, the true ground states $|\psi_0\rangle$ and their exact ground state energy E_0 is obtained via numerical diagonalization of the Hamiltonian. Although reaching small energy deviations

$$\Delta E = \frac{|E - E_0|}{N} \qquad \text{(Equation 9)}$$

indicates that the algorithm has converged to the ground state, we also consider the state overlap with the exact ground state, i.e., the quantum infidelity

$$1 - F = 1 - |\langle \psi_\theta | \psi_0 \rangle|, \qquad \text{(Equation 10)}$$

to verify the accuracy of the obtained state representation. We train for a large number of iterations (typically 1500) keeping track of energy deviations and infidelities.

Transverse-field Ising model (TFIM)

We test the above algorithm on the 1D TFIM whose Hamiltonian consists of nearest-neighbor Ising couplings and a homogeneous transverse field,

$$H_{\text{TFIM}} = -J \sum_{\langle i,j \rangle} \sigma_z^i \sigma_z^j - h \sum_{i=1}^{N} \sigma_x^i, \qquad \text{(Equation 11)}$$

where J is the interaction strength, h is the strength of the external field and $\langle i, j\rangle$ signifies nearest neighbor pairs. Periodic boundary conditions are used such that there is an interaction between spin 1 and spin N. Furthermore, we consider ferromagnetic interactions where $J > 0$ such that alignment of neighboring spins leads to a lower energy. In this case the Hamiltonian of the TFIM in the z-basis is stoquastic (Bravyi et al., 2008).

In the thermodynamic limit the TFIM features a quantum phase transition at the critical point $J = h$ which separates the ordered phase ($h < J$) where the energy is dominated by the spin-spin interactions $\sigma_z^i \sigma_z^{i+1}$ from the disordered phase ($h > J$) where spins increasingly align with the x-axis because of the influence of the external field σ_x^i.

Thus, the two relevant observables are the magnetization in x-direction

$$\langle \sigma_x \rangle = \frac{\sum_i \langle \sigma_x^i \rangle}{N} \qquad \text{(Equation 12)}$$

and the two-point zz-correlation function

$$C_{zz}(d) = \frac{\sum_i \langle \sigma_z^i \sigma_z^{i+d} \rangle}{N}, \qquad \text{(Equation 13)}$$

where d is the distance between spins.

Spin-spin correlations generically fall off exponentially, $C_{zz}(d) \simeq C_0(h)\exp(-d/\xi_{zz}(h))$ whereas in the vicinity of the critical point this dependence turns into a power law (Karl et al., 2017). Thus the correlation length ξ_{zz} diverges at the critical point indicating the phase transition point. Because we are dealing with finite systems ($N \lesssim 10$) the phase transition point is shifted and the correlation length stays finite, but becomes maximal there. We also note that in the ferromagnetic phase the ground state is a superposition between two components that are strongly z-magnetized in either direction, with an energy gap between symmetric and anti-symmetric superposition that vanishes in the limit of $h/J \rightarrow 0$. Physically, this leads to spontaneous symmetry breaking in the ferromagnetic phase. Interestingly, our SNN approximation will show an analogous symmetry breaking effect.

PERFORMANCE

Ising phase transition

As described above, we trained a generative model using the neuromorphic platform BrainScaleS-2 to represent ground states of the TFIM for a spin chain of size $N = 8$ at various transversal field strengths $h \in \{0.1, 0.5, 0.9, 1.0, 1.25, 5, 10\}$. The observables shown in Figure 3 have been obtained through sampling from the learned neuromorphic quantum states.

Overall, we observe very good agreement with the exact solutions for both magnetization (Figure 3A) and zz-correlations (Figure 3C). Interestingly, the correlation length systematically deviates for field strengths deeper in the ferromagnetic regime. As we will demonstrate, this happens because of symmetry breaking during the learning process.

In Figure 3B the spin-spin correlations in z-direction C_{zz} are shown as a function of distance d. The correlation lengths ξ_{zz} are extracted by fitting the data points of each field strength with the following function (shown as dotted lines),

$$\widehat{C}_{zz}(d) = A \exp(-d/\xi_{zz}) + B \qquad \text{(Equation 14)}$$

where the additional parameters A and B account for finite-size effects. The fit parameters ξ_{zz} and their standard deviations are shown in Figure 3C together with the corresponding theoretical values (solid line). We observe that the correlation length has a maximum at $h/J \approx 1.25$ marking the phase transition point and closely matching the theoretical prediction. Although for $h/J \geq 0.9$ the results agree well with the exact values for both observables, $\langle \sigma_x \rangle$ and ξ_{zz}, at $h/J \in \{0.1, 0.5\}$ the correlation length is significantly underestimated.

To illustrate the origin of this deviation, we show the probabilities for finding the system in a state with z-magnetization m (half of the difference between the number of up- and down-spins in $\boldsymbol{v}$) in Figure 3D. This reveals that instead of the symmetrical ground state distribution which is learned correctly for $h \geq 0.9$, the symmetry is broken for low field values. For $h = 0.1$ it is shown that two different ground states with all spins up or down can be reached (see Methods E: Supplementary analysis of symmetry breaking for more details). The average of these two distributions (dotted line) is a good approximation to the symmetric distribution.

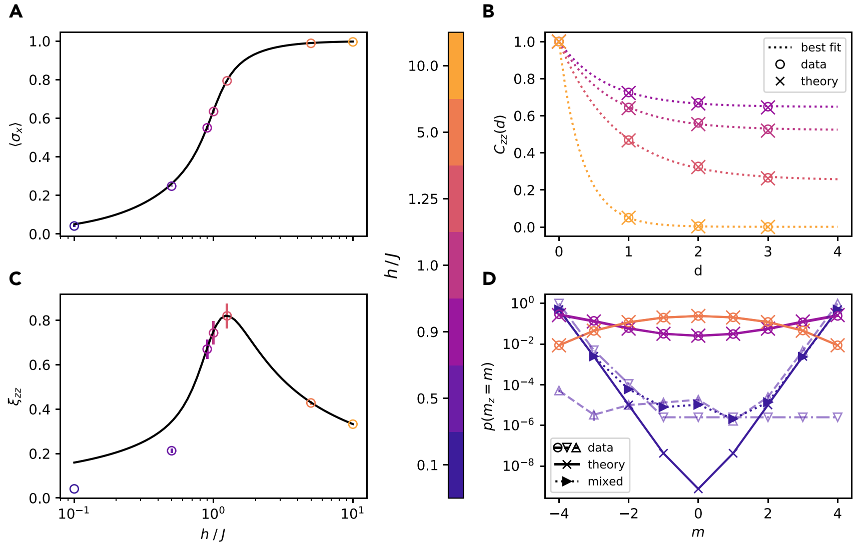


Figure 3. TFIM ground-state learning

(A) Average x-magnetization $\langle\sigma_x\rangle$ of an $N = 8$ spin Ising system for different external fields h/J. Measurement errors are smaller than the marker size and not shown. The marker colors identify the field strength h/J in all panels.

(B) zz-correlation for different external fields $h/J \in \{0.1, 0.5, 0.9, 5.0\}$ as function of spin distance d. An exponential fit (Equation (14), dotted line) was applied to the data (circles, errors not shown like in (A)) which are in agreement with theory (crosses).

(C) As (A) but for the zz-correlation length ξ_{zz}. Shown error bars are standard deviations over the last 200 training epochs. Deviations are observable for small h.

(D) Distribution of observed z-magnetization values for different h. Although for $h \in \{1, 5\}$ symmetric distributions are learned, one observes spontaneous symmetry breaking for the lower field value $h = 0.1$. In this case, whether the $m > 0$ or the $m < 0$ component of the ground state is found depends on the choice of initial parameters of the network. Averaging over opposite initialisations ($\triangle$, ∇) results in a good approximation ($\triangleright$, mixed). The remaining apparent difference is because of the limited number of samples (where we replaced 0 entries by 10^{-6}). Statistical variations are too small to be resolved, note also the logarithmic y-axis.

Such spontaneous symmetry breaking happens physically whenever the ground state of the system is (near-) degenerate, because any small perturbation of the system will break the symmetry and collapse the macroscopic superposition into one of its components. That is the exact ground state becomes harder to prepare for $h \to 0$ as it then increasingly approaches a superposition of the two extreme configurations $|\downarrow\rangle^{\otimes N}$ and $|\uparrow\rangle^{\otimes N}$. In a way, we see the same behavior reproduced by the neuromorphic device. This is because in terms of SNN dynamics, such a distribution requires both highly synchronous activity and synchronous inactivity. Achieving such a behavior requires distributions with strong local minima, making it hard for any MCMC method to escape. This so-called mixing problem already manifested itself in the increased need for samples at $h/J = 0.9$ in order to well represent the symmetric ground state. The points $h/J \in \{0.1, 0.5\}$, are even deeper in the ferromagnetic phase which made learning these highly entangled states prohibitively hard with our static stochasticity system.

Dependence on system size

In order to assess the scalability of our approach we studied its performance for different sizes of the quantum system. In the experiment shown in Figure 4 the number of spins N is increased from $N = 3$ to $N = 10$ for the critical point $h/J = 1$. For details of the used network parameters and sample sizes, see Methods F: Details on the choice of network parameters. Note that the SNN has less parameters than the number of wave function coefficients for $N = 9$ and $N = 10$.

Overall, a quantum fidelity greater than 99% can be achieved and even up to 99.9% for systems $N \leq 6$ (Figure 4B). Because the fidelity imposes an upper bound on the errors of any possible expectation values, good agreement of the learned observables is guaranteed.

Figures 4C and 4D show exemplary learning curves of energy error and fidelity as a function of the training iteration. Although the learning curves converge quickly for small system sizes, it takes progressively longer

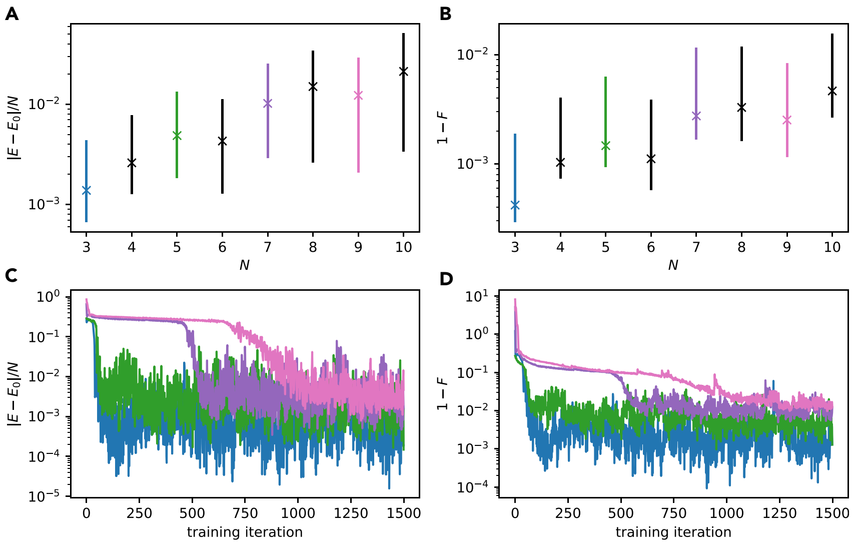


Figure 4. System-size dependence

Performance at $h/J = 1$ as a function of system size.

(A and B) Relative energy mismatch and infidelity between the learned and exact ground state increases at fixed number of hidden units. We report median values and the 15- and 85-percentiles over the last 200 iterations as error bars.

(C and D) Evolution of the approximation quality during learning.

to reach good metrics for larger systems, with intermediate regimes of very slow improvement. This behavior is well-studied in the machine learning literature and related to the high number of saddle-points in the parameter space (Dauphin et al., 2014). It should be noted that these plateaus are not observed for $h/J = 10$, where the ground state distribution becomes more uniform, which is easy to reach by gradient descent independent of the initial conditions.

Limitations

For system sizes above $N = 10$, we observed a significant drop in the performance of our neuromorphic implementation after learning. One reason for this is that we are limited to a purely sample-based estimate when calculating gradients. Estimating expressions like $E_{\mathbf{v}}^{\mathrm{loc}}$ (Equation 8) requires an approximation of the full distribution $p_\theta(\mathbf{v})$. Depending on the nature of the sampled distribution, MCMC methods need a certain number of samples to reach a given precision; this number scales linearly with the size of the relevant sample space which, in the worst case, scales exponentially with the number of physical spins (Speagle, 2019). This is further discussed and illustrated in Methods G: Comparing with CPU-based implementation. This issue could be overcome by explicitly computing the factors $p_\theta(\mathbf{v}')/p_\theta(\mathbf{v})$ (cf. Equation 17) for a given set of samples $\boldsymbol{v}$ and physical network parameters, as discussed in Section discussion.

The second source of error relates to the properties of the neuromorphic substrate used. In the following, we thoroughly study the impact of substrate induced limitations on the performance of our method. In particular, we consider (1) limited hidden layer size, (2) limited network parameter range, (3) finite network parameter resolution, (4) non-optimal choice of the learning rate, and (5) deviations of the substrate from the theoretically assumed dynamics.

Hidden layer size

In order to assess the required number of hidden units for a good variational representation depending on the system size, we have performed a grid search over $(N, N_h) = (\{3, \ldots, 8\}, \{5, 10, 20\})$ drawing $N_{\mathrm{sample}} = 2 \cdot 10^5$ samples for 1500 training iterations each.

The results are shown in Figures 5A and 5B in terms of median energy error per spin and infidelity of the state representation averaged over the last 200 training iterations. Although $N_h = 5$ (red line) is sufficient to accurately describe the systems for $N < 6$, both energy error and infidelity increase sharply for larger N.

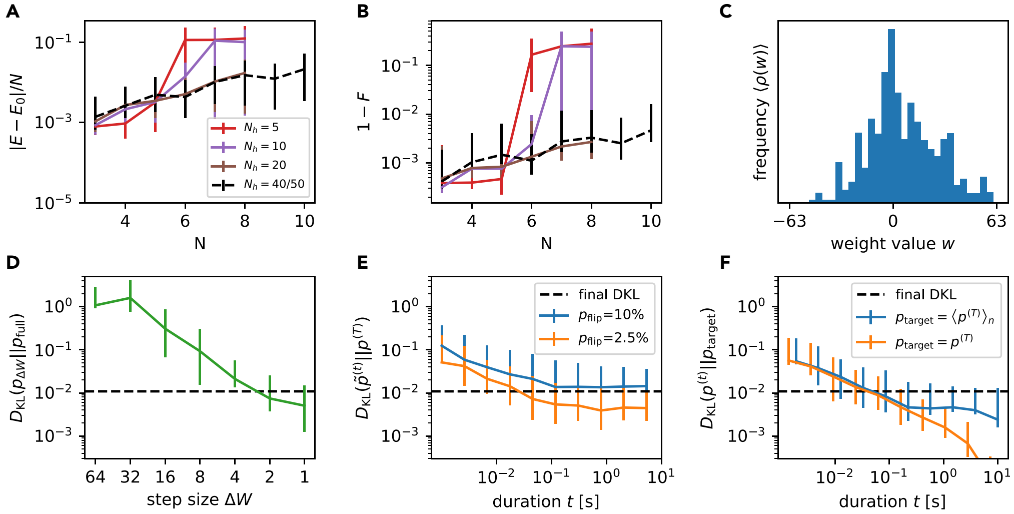


Figure 5. Analysis of hardware limitations

(A and B) Approximation quality as a function of the hidden layer size, analogous to Figures 4A and 4B. Small hidden layers can limit the fidelity of the learned state, especially for larger systems. For the system sizes used here ($N < 10$) hidden layer sizes of $N_h = 20$ have proven to be sufficient. We report median values and the 15- and 85-percentiles over the last 200 iterations as error bars (same as Figures 4A and 4B).

(C) Weight distribution accumulated over the final 200 epochs for $N = 8$, $N_h = 20$. The weights are not clipped significantly by the limited range $-63 \leq w \leq +63$.

(D) Effect of the weight resolution: D_{KL} between the full 7-bit distribution and a coarse grained one as a function of the smallest possible weight step Δw. For comparison: A successfully trained system with $N = 8$ and $N_h = 20$ reached a final $D_{\mathrm{KL}} \approx 10^{-2}$ (dashed horizontal line, also in (E) and (F)). We report median and 15- and 85-percentiles over 10 repetitions as error bars.

(E) Comparison between a reference distribution $p^{(T)}$ and a distribution perturbed by a "pseudo weight update". We show the D_{KL} between these distributions as a function of hardware execution time $t < T$ used for sampling the perturbed distributions $\tilde{p}^{(t)}$. We report median and 15- and 85-percentiles over 30 repetitions as error bars.

(F) Convergence behavior for a static configuration: Comparing to the final distribution of a single run (orange) we observe the ideal $1/N_{\mathrm{sample}}$ behavior. Convergence towards an average distribution $\langle p^{(T)} \rangle_n$ over multiple runs stops at a sampling time of about 0.2 s (blue). Note, for visibility reasons we plot alternative times for the different experiments. We report median and 15- and 85-percentiles over 30 repetitions as error bars (same as in E).

Increasing the number of hidden neurons to $N_h = 10$ (purple line) allows us to obtain accurate ground state representations up to $N = 6$ with $F > 99\%$, while $N_h \geq 20$ (brown line) is required for $N \in \{7, 8\}$.

One might expect that using more hidden units could decrease the slope of the curve further, also bringing the large systems above 99.9% fidelity. However, comparing with $N_h = 40$ (black dashed line, from Figures 4A and 4B), there is no significant difference in either energy error or fidelity, suggesting that model capacity is not the dominating limitation. Although the system is indeed overparameterized in the sense that there are more variational parameters than wave function coefficients, the physical network parameters can only be controlled with finite precision, and within a finite range. These bounds may limit the representational power of the ansatz, as we discuss in the next two sections.

Weight range

The strongest realizable weights on BSS-2 are represented by the digital values $w_{\mathrm{max}} = \pm 63$. Figure 5C shows a typical weight distribution accumulated over the last 200 training iterations (for $N = 8$ and $N_h = 20$). The distribution is peaked around zero with roughly symmetrical tails. However, there is no significant occupancy of the outermost weight values, hence clipping beyond the edges of ± 63 should have no effect. We conclude that the chosen weight range is sufficient and does not restrict the achievable representation accuracy.

Weight resolution

On the BSS-2 system, the synaptic connections are implemented by two 6-bit configurable circuits, one for the excitatory ($w_{ij} > 0$) and one for the inhibitory ($w_{ij} < 0$) part of the synaptic connectome. We therefore used two physical synapses to form a logical synapse which gives an additional bit for the sign (see Methods A: Description of the BrainScaleS-2 ASIC for details).

A lower parameter resolution leads to a more coarse-grained space of representable distributions. To assess whether this has a detrimental impact, and hence is a limiting factor for learning performance, we conducted an experiment where we randomly initialized the network weights ($N_v = 8, N_h = 20$) drawing from a uniform distribution $w_{ij} \sim \mathcal{U}(-63, 63)$. The neuron biases were set to the center (i.e., half the maximum spike rate) of their respective activation functions (cf. Figure 1C). We then artificially reduced the resolution of the weights, and thus of the distribution, by defining a grid centered at zero and with a minimum step size $\Delta w > 1$ between two allowed weight values. We compare distributions sampled using the full parameter resolution $p_{\text{full}} = p_{\Delta w = 1}$ to the distributions $p_{\Delta w}$ obtained by rounding the weights to the low resolution grids with step sizes Δw. We quantify the distance between these distributions by the Kullback-Leibler divergence

$$D_{\text{KL}}(p \| q) = -\sum_{v} p(v) \log(p(v)/q(v)) \qquad \text{(Equation 15)}$$

such that $p = p_{\text{full}}$ and $q = p_{\Delta w}$. For every Δw we repeat 10 sampling experiments each of duration $T = 0.1$ s.

As Figure 5D shows, we find a quick decrease in D_{KL} as the step size shrinks, which, however, plateaus for $\Delta W \leq 2$. The achieved D_{KL} for $\Delta w \in \{1, 2\}$ is consistent with the typical final $D_{\text{KL}} \simeq 10^{-2}$ (dashed line) for trained networks of the same size. Therefore, we conclude that the limited parameter precision also does not explain the saturation in the observed learning performance.

Note that the network observed here had ample representational power for the system size (cf. Figure 4). It may be that a more significant effect could be observed for smaller hidden layers. Furthermore, no training was performed in order to isolate the effect of finite weight resolution on the accuracy of the sampled distribution.

Learning rate

In all our ground state learning experiments we used a learning rate decay (see Methods C: Adaptive momentum optimization) to facilitate the descent into minima of the energy landscape. In addition, towards the end of the training the gradients become small thereby also shrinking the weight updates. At late stages of the training we typically observe changes in 2–3% of the individual discrete weights. A potential limitation to the achievable convergence is a still too high learning rate at the end of the training which prevents precise descent into local minima. To test whether this is the case, we perturbed a reference distribution with a ″pseudo update″ and observed the size of the resulting deviation measured by the D_{KL}.

In particular, we again initialized our system with a uniformly random distributed weight matrix $w_{ij} \sim \mathcal{U}(-62, 62)$ and collected samples from it over a period of $T = 10$ s – significantly longer than needed for convergence ≈ 0.1 s. This defined our reference distribution $p^{(T)}$. We then simulated a weight update by changing a fraction p_{flip} of the weight parameters w_{ij} by ± 1 and again sampled from the modified distribution. This defined a perturbed distribution $\tilde{p}^{(T)}$. Using only samples up to some time $t < T$ defined a series of perturbed distributions $\tilde{p}^{(t)}$. In Figure 5E we demonstrate the evolution of the resulting D_{KL} between these perturbed distributions and the reference distribution.

We observe that the D_{KL} (green curve) decreases quickly until around $t = 0.1$ s after which it saturates because of the distortion induced by the random weight changes. $p_{\text{flip}} = 10\%$ was chosen such that the final D_{KL} corresponds to the observed final D_{KL}s during training (dashed horizontal line). On the other hand, for a value of $p_{\text{flip}} = 2.5\%$ we observe that a better approximation is reached. Because we observed 2–3% weight flips per learning update at the late stages of the actual training, this result indicates that the ground state search is not limited by a too large learning rate.

Temporal stability of the substrate

The key feature of the BSS-2 system – and the main catalyst of its speed and efficiency – is the analog nature of its neuro-synaptic dynamics. However, its direct benefits for our approach come with a number of specific challenges that do not appear in digital devices or simulations, such as a certain amount of component diversity, as shown in Figure 1C. Although this particular effect is automatically corrected for during learning, other phenomena are more subtle and difficult to compensate. An immanent property of analog components is the presence of small instabilities and drifts in their parameters (Schemmel et al., 2010; Pfeil et al., 2013; Schmitt and Heyl, 2018). In this section we study the impact of such effects on the sampling and learning performance by conducting long-duration sampling experiments of $T = 10$ s (initialization as in Section weight resolution).

First, we compare the convergence during a single run by measuring $D_{\mathrm{KL}}(\tilde{p}^{(t)} \| p^{(T)})$ (orange line in Figure 5F). Here, by construction, convergence to zero is assured and we see the expected $1/N_{\mathrm{sample}}$ behavior of Monte Carlo sampling. Our aim was to test the reproducibility and stability of the sampling procedure over multiple iterations and reconfigurations for a fixed parameter set θ. In a second experiment, we therefore repeated the sampling procedure for $T = 10$ s for $n = 30$ times and averaged the resulting distributions. We then compared the observed distributions $p^{(t)}$ to the average target distribution $p_{\mathrm{target}} = \langle p^{(T)} \rangle_n$. Initially, the D_{KL} gradually decreased as more samples were gathered (Figure 5F). However, beyond 200 ms the D_{KL} saturated. This shows that even a repeated experiment with the exact same configuration of network parameters θ samples from a slightly different distribution than the original one p_θ. This, in turn, indicates that the parameters of the physical system do not stay constant over the duration of an entire experiment.

The timescale of variability observed above is significantly shorter than the total duration of both training and evaluation, each of which covered at least 200 epochs of 100 ms. We thus conclude that the temporal variability of the analog parameters represents the main limiting factor for the fidelity of our approach on BSS-2. For larger system sizes, where more samples are required to obtain precise gradient estimates, this effect becomes increasingly severe and thus causes the observed drop in the representational power of our neuromorphic implementation. Understanding this limitation points directly to possible mitigation strategies, which we address in the discusion below.

DISCUSSION

In summary, we have presented a demonstration of neuromorphic ground state search for quantum spin systems. We have designed a variational algorithm suitable for implementation in the mixed-signal BSS-2 system which enables fast spike-based sampling in an inherently parallel fashion and independent of the network size. These advantages could provide significant speedups for the emulation of large networks or quantum spin systems. For this reason we have tested the scalability of our approach, thereby expanding previous work by Czischek et al. (2022) from representing small entangled states to larger quantum spin systems of up to $N = 10$ spins. Furthermore, we have analyzed the TFIM phase transition and found excellent agreement with exact solutions. In the ferromagnetic regime we observed symmetry breaking in the SNN activity reflecting the tendency of the quantum spin system toward spontaneous order.

For systems with $N > 10$, the reachable approximation quality decreased sharply. By systematically studying potential limiting factors, we were able to exclude several possible causes of this degradation, namely the limited number of hidden neurons, finite weight range and resolution, as well as non-optimal learning rate. Moreover, we found that the currently available parameter stability on BSS-2 leads to a limited accuracy of gradients and thus represents the main technical obstacle to be overcome for further improving the approximation quality at large system sizes.

A second, algorithmic limitation of our learning scheme is the requirement of the precise knowledge of $p(\boldsymbol{v}')$ for all non-zero observed $p(\boldsymbol{v})$ that are connected by a non-zero $H_{\boldsymbol{v}\boldsymbol{v}'}$. We showed that the effect of this limitation on purely sampling-based methods is independent of the computational substrate as it is shared by CPU-based implementations (see Methods G: Comparing with CPU-based implementation).

Neither the technical nor the algorithmic challenges are fundamental roadblocks for using neurmorphic hardware for variational learning of quantum states and will be addressed in future research. Because BSS-2 was developed as a multi-purpose research system, its capabilities were not optimized for spike-based sampling. Advancements in the development of BSS-2 and other neuromorphic hardware platforms (Roy et al., 2019) will alleviate technical issues and introduce new capabilities and tools. For analog, and in particular accelerated platforms, parameter variability over typical experiment durations of tens to thousands of seconds can be greatly reduced. Furthermore, increasing the system size beyond

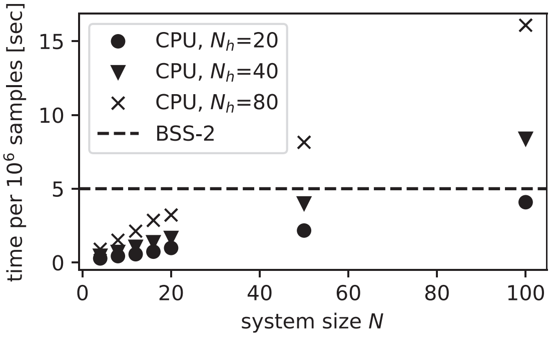


Figure 6. Computation-time scaling

Scaling behavior of the sample generation for different sizes of the physical system and three different sizes of the hidden layer. Because of the physical nature of BSS-2, its emulation time remains constant, whereas simulation time increases linearly for the CPU implementation. Note that while the exact measurement values depend on the choice of CPU and parametrization of BSS-2, the difference in scaling is fundamental.

the neuron number currently available on a single BSS-2 chip will likely require a multi-chip setup with low-latency connections (see, e.g., Schemmel et al., 2010; Thommes et al., 2022, but also Petrovici et al., 2017).

On the other hand, using purely digital neuromorphic chips such as ODIN (Frenkel et al., 2018) or Loihi (Davies et al., 2018) would circumvent the instabilities of an analog system and thus permit scaling to larger quantum system sizes. Although this might come at the cost of losing some of the analog advantages of BSS-2, mainly with respect to speed and energy efficiency, it will likely still outperform more conventional, CPU/GPU-based solutions (Göltz et al., 2021) (see Figure 6 and Methods G: Comparing with CPU-based implementation). In either scenario, improved control and readout of the neuromorphic substrate could also allow the direct calculation of Boltzmann factors from the weight and bias parameters. This would enable the efficient computation of local energies (Equation 8) and thus solve the problem of having to densely sample the visible distribution (Carleo and Troyer, 2017).

For small transverse fields we observed symmetry breaking during the training. The parity symmetry corresponding to a global spin flip required two differently initialized training runs to be reproduced deep in the ferromagnetic phase. The root cause is the near-degeneracy of the two highly synchronous states (all active, all inactive). This corresponds to the well-known mixing problem for which spike-based solutions have been proposed (Leng et al., 2018; Korcsak-Gorzo et al., 2022) which would be amenable to a neuromorphic implementation and should allow a faithful representation of such distributions without the need for re-initialization. From an algorithmic perspective one could also enforce this symmetry by supplementing the generated sample sets with the corresponding spin-flipped configuration for each sample generated by the network. This technique can be employed to enforce any given symmetry of the physical model (Choo et al., 2018; Bukov et al., 2021; Nomura, 2021).

Another promising idea for scalable algorithms is the use of local learning rules that only involve connected neuron pairs because most modern neuromorphic platforms support local on-chip learning. An example for training RBMs with a local learning rule is contrastive divergence (Hinton, 2012), for which an event-driven SNN version has been proposed (Neftci et al., 2014). In addition, such generative networks can be further fine-tuned using error backpropagation, which, in turn, can be approximated by local learning rules (Whittington and Bogacz, 2017; Sacramento et al., 2018; Crafton et al., 2019; Lee et al., 2020; Haider et al., 2021), including spike-based variants already demonstrated on BSS-2 (Billaudelle et al., 2020; Göltz et al., 2021). The question of how to translate these local update schemes to variational ground state learning is left as an important direction for future research.

Finally, algorithmic improvements could be enabled by novel encodings of NQS with SNNs. A straightforward idea for encoding not only the amplitudes, but also phases of the wavefunction would be to use additional output units or even a second network like in (Torlai et al., 2018). Phasor networks represent another possible avenue for encoding complex numbers with SNNs. It was shown that these networks, which consist of resonate-and-fire neurons with complex dynamical variables, can be implemented by integrate-and-fire SNNs and can robustly leverage spike-timing codes (Frady and Sommer, 2019). If successful, these approaches to representing complex values in SNNs could enable the extension of the presented variational method to non-stoquastic systems.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Data and code availability
- METHOD DETAILS
  - Methods A: Description of the BrainScaleS-2 ASIC
  - Methods B: Derivation of the learning rule
  - Methods C: Adaptive momentum optimization
  - Methods D: Leaky integrate-and-fire neurons
  - Methods E: Supplementary analysis of symmetry breaking
  - Methods F: Details on the choice of network parameters
  - Methods G: Comparing with CPU-based implementation

ACKNOWLEDGMENTS

We thank G. Carleo, S. Czischek and T. Gasenzer for helpful discussions. We thank the Electronic Vision(s) group, in particular S. Billaudelle, B. Cramer, J. Schemmel, E. C. Müller, C. Mauch, Y. Stradmann, P. Spilger, J. Weis, and A. Emmel, for maintaining and providing access to the BrainScaleS-2 system and for technical support. This work is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2181/1 – 390900948 (the Heidelberg STRUCTURES Excellence Cluster), by the DFG – project-ID 273811115 – SFB 1225 (ISOQUANT), by the European Union 7th and Horizon-2020 Framework Programmes and under grant agreements 785907, 945539 (HBP), and by the Manfred Stärk Foundation. This work was partially financed by the Baden-Württemberg Stiftung gGmbH.

AUTHOR CONTRIBUTIONS

RK performed the experiments. AB developed the employed software interface for spike-based sampling in BSS2. RK and AB evaluated the experiment data. MG, MP and AB designed the experiments. MG and MP supervised the project. All authors contributed to writing the manuscript.

DECLARATION OF INTERESTS

The authors declare they have no competing interests.

Received: January 13, 2022
Revised: May 5, 2022
Accepted: June 28, 2022
Published: August 19, 2022

REFERENCES

Ackley, D.H., Hinton, G.E., and Sejnowski, T.J. (1985). A learning algorithm for Boltzmann machines. Cognit. Sci. *9*, 147–169.

Avella, A., and .Mancini, F., eds. Strongly correlated systems: theoretical methods. In Springer Series in Solid-State Sciencesed (Springer-Verlag).

Becca, F., and Sorella, S. (2017). Quantum Monte Carlo Approaches for Correlated Systems (Cambridge University Press).

Billaudelle, S., Stradmann, Y., Schreiber, K., Cramer, B., Baumbach, A., Dold, D., Göltz, J., Kungl, A.F., Wunderlich, T.C., Hartel, A., et al. (2020). Versatile emulation of spiking neural networks on an accelerated neuromorphic substrate. In 2020 IEEE International Symposium on Circuits and Systems (ISCAS) (IEEE), pp. 1–5.

Bravyi, S., Divincenzo, D.P., Oliveira, R., and Terhal, B.M. (2008). The complexity of stoquastic local Hamiltonian problems. Quantum Inf. Comput. *8*, 361–385.

Bukov, M., Schmitt, M., and Dupont, M. (2021). Learning the ground state of a non-stoquastic quantum Hamiltonian in a rugged neural network landscape. Sci. Post Phys. *10*, 147.

Carleo, G., and Troyer, M. (2017). Solving the quantum many-body problem with artificial neural networks. Science *355*, 602–606.

Carrasquilla, J., Torlai, G., Melko, R.G., and Aolita, L. (2019). Reconstructing quantum states with generative models. Nat. Mach. Intell. *1*, 155–161.

Carrasquilla, J. (2020). Machine learning for quantum matter. Adv. Phys. X *5*, 1797528.

Choo, K., Carleo, G., Regnault, N., and Neupert, T. (2018). Symmetries and many-body excitations with neural-network quantum states. Phys. Rev. Lett. *121*, 167204.

Crafton, B., Parihar, A., Gebhardt, E., and Raychowdhury, A. (2019). Direct feedback alignment with sparse connections for local learning. Front. Neurosci. *13*, 525.

Czischek, S., Baumbach, A., Billaudelle, S., Cramer, B., Kades, L., Pawlowski, J.M., Oberthaler, M.K., Schemmel, J., Petrovici, M.A., Gasenzer, T., and Gärttner, M. (2022). Spiking neuromorphic chip learns entangled quantum states. J. Neurol. *269*, 39–43.

Czischek, S., Gärttner, M., and Gasenzer, T. (2018). Quenches near Ising quantum criticality as a challenge for artificial neural networks. Phys. Rev. B *98*, 024311.

Dauphin, Y.N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In Advances in Neural Information Processing Systems, *27*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds. (Curran Associates, Inc).

Davies, M., Srinivasa, N., Lin, T.-H., Chinya, G., Cao, Y., Choday, S.H., Dimou, G., Joshi, P., Imam, N., Jain, S., et al. (2018). Loihi: a neuromorphic many core processor with on-chip learning. IEEE Micro *38*, 82–99.

Dold, D., Bytschok, I., Kungl, A.F., Baumbach, A., Breitwieser, O., Senn, W., Schemmel, J., Meier, K., and Petrovici, M.A. (2019). Stochasticity from function–why the bayesian brain may need no noise. Neural Netw. *119*, 200–213.

Frady, E.P., and Sommer, F.T. (2019). Robust computation with rhythmic spike patterns. Proc. Natl. Acad. Sci. USA *116*, 18050–18059.

Frenkel, C., Lefebvre, M., Legat, J.-D., and Bol, D. (2018). A 0.086-mm^2 12.7-pj/sop 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm cmos. IEEE Trans. Biomed. Circuits Syst. *13*, 145.

Geman, S., and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell. *6*, 721–741.

Gerstner, W., and Wulfram, K. (2002). Spiking Neuron Models: Single Neurons, Populations, Plasticity (Cambridge University Press).

Göltz, J., Kriener, L., Baumbach, A., Billaudelle, S., Breitwieser, O., Cramer, B., Dold, D., Kungl, A.F., Senn, W., Schemmel, J., et al. (2021). Fast and energy-efficient neuromorphic deep learning with first-spike times. Nat. Mach. Intell. *3*, 823–835.

Haider, P., Ellenberger, B., Kriener, L., Jordan, J., Senn, W., and Petrovici, M.A. (2021). Latent equilibrium: a unified learning theory for arbitrarily fast computation with arbitrarily slow neurons. In Advances in Neural Information Processing Systems, *34*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J.W. Vaughan, eds. (Curran Associates, Inc), pp. 17839–17851.

Hartmann, M.J., and Carleo, G. (2019). Neural-network approach to dissipative quantum many-body dynamics. Phys. Rev. Lett. *122*, 250502.

Hinton, G.E., Dayan, P., Frey, B.J., and Neal, R.M. (1995). The "wake-sleep" algorithm for unsupervised neural networks. Science *268*, 1158–1161.

Hinton, G.E. (2012). A practical guide to training restricted Boltzmann machines. In Neural networks: Tricks of the trade (Springer), pp. 599–619.

Jia, Z., Yi, B., Zhai, R., Wu, Y., Guo, G., and Guo, G. (2019). Quantum neural network states: a brief review of methods and applications. Adv. Quantum Technol. *2*, 1800077.

Karl, M., Cakir, H., Halimeh, J.C., Oberthaler, M.K., Kastner, M., and Gasenzer, T. (2017). Universal equilibrium scaling functions at short times after a quench. Phys. Rev. E *96*, 022110.

Kingma, D.P., and Ba, J. (2014). Adam: a method for stochastic optimization. Preprint at arXiv. https://doi.org/10.48550/arXiv.1412.6980.

Korcsak-Gorzo, A., Müller, M.G., Baumbach, A., Leng, L., Breitwieser, O.J., van Albada, S.J., Senn, W., Meier, K., Legenstein, R., and Petrovici, M.A. (2022). Cortical oscillations support sampling-based computations in spiking neural networks. PLoS Comput. Biol. *18*, e1009753.

Kungl, A.F., Schmitt, S., Klähn, J., Müller, P., Baumbach, A., Dold, D., Kugele, A., Müller, E., Koke, C., Kleider, M., et al. (2019). Accelerated physical emulation of bayesian inference in spiking neural networks. Front. Neurosci. *13*, 1201.

Lee, J., Zhang, R., Zhang, W., Liu, Y., and Li, P. (2020). Spike-train level direct feedback alignment: sidestepping backpropagation for on-chip training of spiking neural nets. Front. Neurosci. *14*, 143.

Leng, L., Martel, R., Breitwieser, O., Bytschok, I., Senn, W., Schemmel, J., Meier, K., and Petrovici, M.A. (2018). Spiking neurons with short-term synaptic plasticity form superior generative networks. Sci. Rep. *8*, 10651.

Long, P.M., and Servedio, R.A. (2010). Restricted Boltzmann machines are hard to approximately evaluate or simulate. In Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10 (Omnipress), pp. 703–710.

Melko, R.G., Carleo, G., Carrasquilla, J., and Cirac, J.I. (2019). Restricted Boltzmann machines in quantum physics. Nat. Phys. *15*, 887–892.

Nagy, A., and Savona, V. (2019). Variational quantum Monte Carlo method with a neural-network ansatz for open quantum systems. Phys. Rev. Lett. *122*, 250501.

Neftci, E., Das, S., Pedroni, B., Kreutz-Delgado, K., and Cauwenberghs, G. (2014). Event-driven contrastive divergence for spiking neuromorphic systems. Front. Neurosci. *7*, 272.

Nomura, Y. (2021). Helping restricted Boltzmann machines with quantum-state representation by restoring symmetry. J. Phys. Condens. Matter *33*, 174003.

Pehle, C., Billaudelle, S., Cramer, B., Kaiser, J., Schreiber, K., Stradmann, Y., Weis, J., Leibfried, A., Müller, E., and Schemmel, J. (2022). The brainscales-2 accelerated neuromorphic system with hybrid plasticity. Front. Neurosci. *16*, 795876. https://doi.org/10.3389/fnins.2022.795876.

Petrovici, M.A., Schroeder, A., Breitwieser, O., Grübl, A., Schemmel, J., and Meier, K. (2017). Robustness from structure: inference with hierarchical spiking networks on analog neuromorphic hardware. In 2017 International Joint Conference on Neural Networks (IJCNN), pp. 2209–2216.

Petrovici, M.A., Vogginger, B., Müller, P., Breitwieser, O., Lundqvist, M., Muller, L., Ehrlich, M., Destexhe, A., Lansner, A., Schüffny, R., et al. (2014). Characterization and compensation of network-level anomalies in mixed-signal neuromorphicmodeling platforms. PLoS One *9*, e108590.

Petrovici, M.A. (2016). In Form Versus Function: Theory and Models for Neuronal Substrates (Springer International Publishing).

Petrovici, M.A., Bill, J., Bytschok, I., Schemmel, J., and Meier, K. (2016). Stochastic inference with spiking neurons in the high-conductance state. Phys. Rev. E *94*, 042312.

Pfeil, T., Grübl, A., Jeltsch, S., Müller, E., Müller, P., Petrovici, M.A., Schmuker, M., Brüderle, D., Schemmel, J., and Meier, K. (2013). Six networks on a universal neuromorphic computing substrate. Front. Neurosci. *7*, 11.

Preskill, J. (2018). Quantum computing in the NISQ era and beyond. Quantum *2*, 79.

Reh, M., Schmitt, M., and Gärttner, M. (2021). Time-dependent variational principle for open quantum systems with artificial neural networks. Phys. Rev. Lett. *127*, 230501.

Roy, K., Jaiswal, A., and Panda, P. (2019). Towards spike-based machine intelligence with neuromorphic computing. Nature *575*, 607–617.

Sacramento, J., Ponte Costa, R., Bengio, Y., and Senn, W. (2018). Dendritic cortical microcircuits approximate the backpropagation algorithm. In Advances in Neural Information Processing Systems, *31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds. (Curran Associates, Inc).

Schemmel, J., Brüderle, D., Grübl, A., Hock, M., Meier, K., and Millner, S. (2010). A wafer-scale neuromorphic hardware system for large-scale neural modeling. In Proceedings of the 2010 IEEE International Symposium on Circuits and Systems (ISCAS"10), p. 1947.

Schmitt, M., and Heyl, M. (2018). Quantum dynamics in transverse-field Ising models from classical networks. Sci. Post Phys. *4*, 013.

Schmitt, M., and Heyl, M. (2020). Quantum many-body dynamics in two dimensions with artificial neural networks. Phys. Rev. Lett. *125*, 100503.

Schmitt, S., Klähn, J., Bellec, G., Grübl, A., Güttler, M., Hartel, A., Hartmann, S., Husmann, D., Husmann, K., Jeltsch, S., et al. (2017). Neuromorphic hardware in the loop: training a deep spiking network on the brainscales wafer-scale system. In 2017 International Joint Conference on Neural Networks (IJCNN), pp. 2227–2234.

Schollwöck, U. (2011). The density-matrix renormalization group in the age of matrix product states. Ann. Phys. *326*, 96–192.

Speagle, J.S. (2019). A conceptual introduction to Markov Chain Monte Carlo methods. Preprint at arXiv. https://doi.org/10.48550/arXiv.1909.12313.

Thakur, C.S., Molin, J.L., Cauwenberghs, G., Indiveri, G., Kumar, K., Qiao, N., Schemmel, J., Wang, R., Chicca, E., Hasler, J.O., et al. (2018). Large-scale neuromorphic spiking array processors: a quest to mimic the brain. Front. Neurosci. *12*, 991.

Thommes, T., Bordukat, S., Grübl, A., Karasenko, V., Müller, E., and Schemmel, J. (2022). Demonstrating brainscales-2 inter-chip pulse-communication using extoll. In Neuro-Inspired Computational Elements Conference, NICE 2022 (Association for Computing Machinery), pp. 98–100.

Torlai, G., and Melko, R.G. (2020). Machine-learning quantum states in the NISQ era. Annu. Rev. Condens. Matter Phys. *11*, 325–344.

Torlai, G., Mazzola, G., Carrasquilla, J., Troyer, M., Melko, R., and Carleo, G. (2018). Neural-network quantum state tomography. Nat. Phys. *14*, 447–450.

Whittington, J.C.R., and Bogacz, R. (2017). An approximation of the error backpropagationalgorithm in a predictive coding network with local hebbian synaptic plasticity. Neural Comput. *29*, 1229–1262.

Zhou, X., Lee, W.-S., Imada, M., Trivedi, N., Phillips, P., Kee, H.-Y., Törmä, P., and Eremets, M. (2021). High-temperature superconductivity. Nat. Rev. Phys. *3*, 462–465.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
Spike-based inference	Petrovici et al. (2016) (PRE)	
ADAM optimizer	D. P. Kingma and Ba (2014) (arXiv)	
Other		
BrainScaleS-2	Pehle et al. (2022)	

RESOURCE AVAILABILITY

Lead contact

Additional information: Further information and requests for resources should be directed to, Andreas Baumbach (andreas.baumbach@kip.uni-heidelberg.de).

Data and code availability

- Data availability: Data for all figures is available at GitHub repository.
- Code availability: Reproduction of the actual experiments requires access to BSS-2. The software is available, in principle, at GitHub repository.

METHOD DETAILS

Methods A: Description of the BrainScaleS-2 ASIC

The BSS-2 application-specific integrated circuit (ASIC) features 512 neuron circuits, each capable of emulating the adaptive exponential integrate-and-fire neuron model. With appropriate parametrization, this reduces to the LIF model required by our approach (see Methods D: Leaky integrate-and-fire neurons and Gerstner et al., 2002 for details). These single compartments can be wired to resemble structured neurons. An on-chip analog parameter memory as well as integrated static random-access memory (SRAM) cells allow the individual configuration of each neuron. Each neuron integrates input from 256 dedicated synapses, which carry a 6-bit weight. Synapses can either be exclusively excitatory or exclusively inhibitory. However, combining two neuron circuits to one logical neuron allows us to implement both types of connections between all 256 pairs of such logical neurons on a chip.

This analog core is accompanied by supporting logic, including circuitry for communication and configuration (Pehle et al., 2022). In particular, there is circuitry for providing on-chip high-frequency Poisson spike sources. A routing module allows mixing of these spikes with external stimuli and recurrent events.

BSS-2 also comes with two general purpose embedded custom processors for implementing on-chip plasticity. Future work could make use of these plasticity processing units to realize an on-chip implementation of our training algorithm.

The analog nature of the circuitry results in a slight heterogeneity between different neurons. We compensate for this by configuring the single circuits individually in a way that the resulting logical neuron obeys the desired set of neuron parameters (time constants, etc.). In particular, we choose a small membrane time constant $\tau_m = \frac{C_m}{g_l} \approx 0.5\ \mu\text{s}$ and comparatively large synaptic and refractory time constants $\tau_{\text{syn}} = \tau_{\text{ref}} \approx 10\ \mu\text{s}$.

Each logical synaptic weight is implemented by two 6-bit circuits (one for excitatory weights $w_{ij} > 0$ one for inhibitory weights $w_{ij} < 0$). Biases are set directly using the 10-bit leak potential parameter (V_l in Equation 1). Because of the circuit design we use only a part of the available settings as can be seen in Figure 1C where the domain of the activation functions is restricted to a dynamic range equivalent to about 8-bit. Furthermore, because of the digital-to-analog conversion of these parameters, we have observed a reduction in the resolution of the corresponding membrane potentials by one (least significant) bit.

Methods B: Derivation of the learning rule

We calculate the derivative of the variational energy with respect to a weight W_{ij} of the network assuming a stoquastic Hamiltonian H and the normalized state representation $|\psi\rangle = \sum_{\mathbf{v}} \sqrt{p(\mathbf{v})} |\mathbf{v}\rangle$:

$$\partial_{W_{ij}} E_\theta = \partial_{W_{ij}} \sum_{\mathbf{v}\mathbf{v}'} \sqrt{p(\mathbf{v})p(\mathbf{v}')} H_{\mathbf{v}\mathbf{v}'} \tag{Equation 16}$$

$$= \sum_{\mathbf{v}\mathbf{v}'} \sqrt{\frac{p(\mathbf{v}')}{p(\mathbf{v})}} H_{\mathbf{v}\mathbf{v}'} \partial_{W_{ij}} p(\mathbf{v}) \tag{Equation 17}$$

$$= \sum_{\mathbf{v}\mathbf{v}'} \sqrt{\frac{p(\mathbf{v}')}{p(\mathbf{v})}} H_{\mathbf{v}\mathbf{v}'} \left(\sum_{h} z_i z_j p(\mathbf{v}') - p(\mathbf{v}) \sum_{z'} z'_i z'_j p(\mathbf{z}') \right) \tag{Equation 18}$$

$$= \sum_{\mathbf{v}h} E^{\text{loc}}_{\mathbf{v}} z_i z_j p(\mathbf{z}) - \sum_{\mathbf{v}} E^{\text{loc}}_{\mathbf{v}} p(\mathbf{v}) \sum_{z'} z'_i z'_j p(\mathbf{z}') \tag{Equation 19}$$

$$= \sum_{z} \left(E^{\text{loc}}_{\mathbf{v}} - E_\theta \right) z_i z_j p(\mathbf{z}) \tag{Equation 20}$$

$$= \left\langle \left(E^{\text{loc}}_{\mathbf{v}} - E_\theta \right) z_i z_j \right\rangle_{p(\mathbf{z})} . \tag{Equation 21}$$

From Equations (16 and 17) we have used the symmetry of the Hamiltonian. In Equation (19) the local energy $E^{\text{loc}}_{\mathbf{v}} = \sum_{\mathbf{v}'} H_{\mathbf{v}\mathbf{v}'} \sqrt{p(\mathbf{v}')} / \sqrt{p(\mathbf{v})}$ is introduced and the variational energy appears in Equation (20) because of the relation $E_\theta = \sum_{\mathbf{v}} E^{\text{loc}}_{\mathbf{v}} p(\mathbf{v})$.

To deal with the numerical problem of vanishing entries in $p_\theta(\mathbf{v})$ a small parameter ε is added to it, essentially introducing a bias toward a uniform distribution. The local energy thus reads $E^{\text{loc}}_{\mathbf{v}} = \sum_{\mathbf{v}} H_{\mathbf{v}\mathbf{v}'} \sqrt{p_{\mathbf{v}'} + \varepsilon} / \sqrt{p_{\mathbf{v}} + \varepsilon}$ where $\varepsilon = 10^{-12}$ was used throughout.

With the above derivation the gradient of the BM, $\nabla_\theta E_\theta \equiv \Delta\theta_{\text{BM}}$, can be estimated as sample average. For the learning scheme we assume that the variational energy gradient with respect to the BSS-2 hardware parameters, $\Delta\theta_{\text{BSS}}$, is well approximated by the analogous computation over hardware samples.

Methods C: Adaptive momentum optimization

Because the gradient only provides local guidance, it is advisable to scale its components according to the roughness of the cost landscape. An adaptive step size decay probes the cost surface at increasing resolution as the training progress and bounds the number of steps that need to be computed to reach convergence. We typically employed an exponentially decaying step size $\alpha(t+1) = \alpha(t)\gamma_{\text{lr}}$ such that $1/(1-\gamma_{\text{lr}})$ sets a timescale of required optimization steps. We found the values $\alpha(1) = 1$, $\gamma_{\text{lr}} = 0.999$ to work well in practice.

In addition to the fixed step size decay, we employed the ADAM scheme (Kingma and Ba, 2014) which combines momentum with an adaptive learning rate which is chosen for each network parameter individually. It is a first-order method that estimates mean, $m(t)$, and variance, $v(t)$, of the gradient by exponential running averages with respective decay rates β_1 and β_2:

$$m(t+1) \leftarrow \frac{\beta_1}{1-\beta_1^t} m(t) + \frac{1-\beta_1}{1-\beta_1^t} \Delta\theta_{\text{BSS}}(t) \tag{Equation 22}$$

$$v(t+1) \leftarrow \frac{\beta_2}{1-\beta_2^t} v(t) + \frac{1-\beta_2}{1-\beta_2^t} \Delta\theta^2_{\text{BSS}}(t) \tag{Equation 23}$$

where$\Delta\theta^2_{\text{BSS}}(t)$ is the component-wise square of the gradient.

The parameters are updated according to the inverted relative error of the gradient where $m(t)$ acts as a momentum and $v(t)$ modifies the learning rate

$$\theta_{\text{BSS}}(t+1) \leftarrow \theta_{\text{BSS}}(t) - \eta(t) \frac{m(t)}{\sqrt{v(t)} + \varepsilon_{\text{ADAM}}} . \tag{Equation 24}$$

The small parameter $\varepsilon_{\mathrm{ADAM}}$ is required for regularization purposes. Because$|\Delta\theta(t)/\sqrt{v(t)}| \leq 1$ the update implicitly adapts the step sizes based on the signal-to-noise ratio of the derivatives. The canonical hyper-parameters for ADAM are used: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon_{\mathrm{ADAM}} = 10^{-8}$.

Methods D: Leaky integrate-and-fire neurons

The LIF neuron model belongs to the family of continuous spiking neuron models (Gerstner et al., 2002). The neuron's membrane is modeled as a capacitor with capacitance C_m. It can be charged by the synaptic current stimulus $I^{\mathrm{syn}}(t)$ whereas it is constantly discharged across a leak conductance g_l.

According to Kirchhoff's laws the voltage u across the capacitance is described by

$$C_{\mathrm{m}} \frac{\mathrm{d}u(t)}{\mathrm{d}t} = g_{\mathrm{l}}(V_{\mathrm{l}} - u(t)) + I^{\mathrm{syn}}(t)\,. \qquad \text{(Equation 25)}$$

The potential V_l plays the role of the resting state which is, in the absence of external input, approached on the timescale of the circuit $\tau_{\mathrm{m}} = C_{\mathrm{m}}/g_{\mathrm{l}}$.

The spike mechanism is triggered when the membrane potential crosses a threshold V_{thresh} from below:

$$u(t_{\mathrm{spike}}) = V_{\mathrm{thresh}} \wedge u'(t_{\mathrm{spike}}) > 0\,. \qquad \text{(Equation 26)}$$

After the spike has been fired, the membrane potential is clamped to a reset value during the absolute refractory period τ_{ref}:

$$u(t_{\mathrm{spike}} \leq t \leq t_{\mathrm{spike}} + \tau_{\mathrm{ref}}) = V_{\mathrm{reset}}\,. \qquad \text{(Equation 27)}$$

BSS-2 implements current-based synapses in which case synaptic weights carry the unit of current. The synaptic input of neuron j is determined by the exponential synaptic kernel $\kappa(t) = \Theta(t)\exp(-t/\tau_{\mathrm{syn}})$ convolved with spike trains of presynaptic neurons $S^i(t) = \sum_{t_s}\delta(t - t_s^i)$:

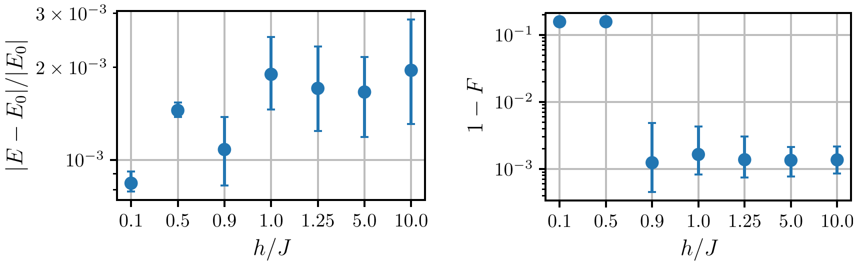
$$I_j^{\mathrm{syn}}(t) = \sum_i w_{ij}(S^i \bigstar \kappa)(t) = \sum_i w_{ij} \sum_{t_s^i} \kappa(t - t_s). \qquad \text{(Equation 28)}$$

The influence of spikes thus decays with the timescale τ_{syn}.

Methods E: Supplementary analysis of symmetry breaking

In the experiments shown in Figure 3 we saw that the symmetry of the ground state was broken for $h \in \{0.1, 0.5\}$ in favor of "spin up" or simultaneous firing of all visible neurons. This bias for the high activity state might be because of the exponential synaptic kernel's influence extending beyond the refractory period.

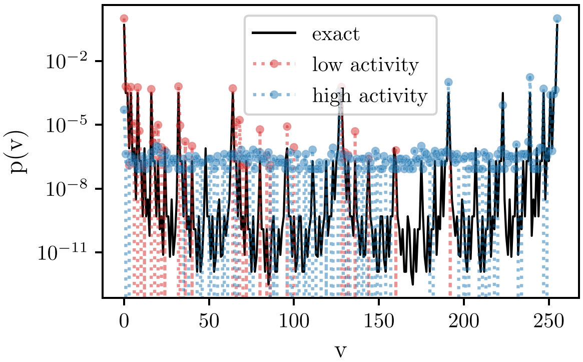
Below Figure shows the energy and infidelity data after training as function of h/J, respectively. The infidelity with the symmetric ground state suddenly jumps to $F \approx 10^{-1}$ for the symmetry broken states. The reason why the observables in Figure 3 were still relatively close to the exact values despite low fidelity is that our minimization objective, the energy expectation value, has the form $E_\theta = -JC_{zz}(d = 1) - h\langle\sigma_x\rangle$. The Hamiltonian is precisely the sum of zz and x terms and thus the symmetry broken states in fact optimize the sum of both observables.



Performance across a quantum phase transition

Relative energy error (left) and infidelity (right) as function of h/J. Note that the field values are not spaced equidistantly. The network parameters used in this figure is as in Figure 3. We report median values and 15- and 85-percentiles over the last 200 iterations as error bars.

By setting an initial negative bias with respect to the neurons' activation functions one can steer the variational algorithm to converge to the opposite symmetry broken state where visible neurons are collectively inhibited. Below Figure compares the learned state for standard bias initialization at the center of the activation functions and for a shift of $\Delta b = -2$ LSB with the symmetrical ground state distribution, confirming this effect.



Spontaneous symmetry breaking in the ordered phase

Symmetry breaking at $h/J = 0.1$. Learned probability distribution over visible neuron configurations $\boldsymbol{v}$ corresponding to basis states of the spin system (dots) compared to the exact ground state distribution (solid). Standard initialization (blue) favors a high activity state, while an initial negative bias offset on all neurons (red) results in final state with low network activity. The network parameters used in this figure is as in Figure 3.

Methods F: Details on the choice of network parameters

Here we specify the network and learning parameters used to produce the data shown in the figures in the main text.

- Figure 3 (Ising phase transition): Each data point used slightly different network topologies and sampling parameters which are summarized in below Table. Note that these parameters were not optimized and most models are overparameterized with respect to the Hilbert space dimension and likely oversampled. For $h = 0.9$ more samples were required in order to adequately learn both modes of the symmetric ground state.

Overview of experiment parameters

h/J	0.1	0.5	0.9	1.0	1.25	5.0	10.0
$N_{\text{sample}}[10^5]$	2	2	4	2	2	2	2
N_h	50	30	40	40	30	20	30
#weights	400	240	320	320	240	160	240
#biases	58	38	48	48	38	28	28

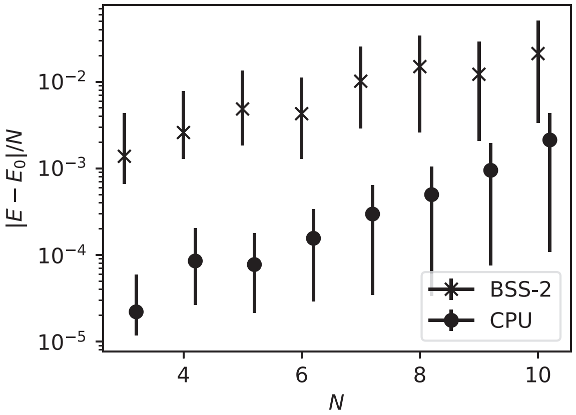
Parameter settings for learning the ground state with $N = 8$ (256 wave function coefficients) at different h/J.

- Figure 4 (system-size dependence): Learning is performed in a network with $N_h = 40$ hidden units and $N_{\text{sample}} = 2 \cdot 10^5$ samples are drawn in each iteration for $N = \{3, \ldots, 8\}$. For $N \in \{9, 10\}$ slightly more hidden units $N_h = 50$ and $N_{\text{sample}} = 4 \cdot 10^5$ samples were used.
- Figure 5C (weight resolution): The uniformly random weights are rounded to subgrids of the full resolution grid $\{-63, -62, \ldots, 0, \ldots, 62, 63\}$ with equidistant steps of sizes $\Delta w \in \{2, 4, 8, 16, 32, 64\}$. We construct the grid starting at zero and counting up to 64 in Δw-steps. Because the maximum possible weight value is 63 we decremented the grid edges from ± 64 to ± 63. Thus, the resulting grids have $128/\Delta w + 1$ possible weight values. Note that $\Delta w = 1$ represents full resolution with 127 weight values.

Methods G: Comparing with CPU-based implementation

Quality

For this comparison, we perform an analogous experiment to the one shown in Figure 4 using a conventional RBM implementation (see below figure, circles) by running the same learning algorithm on a CPU. Gibbs sampling (Geman and Geman, 1984) is used for probabilistic inference of the spin states $p_\theta(\mathbf{v})$ and stochastic estimation of the gradients $\nabla_\theta E_\theta$. Specifically, a network with $N_h = 40$ hidden units is used and $N_{\text{samples}} = 2 \cdot 10^5$ are generated across 10 randomly initialized Markov chains (for better exploration of the state space) in each iteration. The network weights were trained for 10,000 iterations with a learning rate of 0.001.



System-size dependence compared to software models

The energy error of the BSS-2 implementation (crosses) as a function of the system size N (as shown in Figure 4A). The performance of a comparable software RBM (circles) is shown running the same learning scheme with the same number of samples using Gibbs sampling on a CPU. We report median values and the 15- and 85-percentiles over the last 200 iterations as error bars.

As the system size increases we observe an exponential increase of the energy error, albeit at a lower overall error level. This can be explained by the decrease in samples per estimated parameter as the physical system size increases, leading to high variance in the distribution and gradient estimates. Note that unlike BSS-2 the CPU implementation has access to weights and biases with floating-point precision (64-bit, see Methods A: Description of the BrainScaleS-2 ASIC).

This experiment highlights an algorithmic limitation of the employed learning scheme for both CPU and BSS-2 implementations, namely the reliance on a sample estimate of the distribution over spin states $p_\theta(\mathbf{v})$. Implementations in the NQS literature instead compute the local energy associated with a sample $\mathbf{v}$ by exactly calculating the relevant likelihoods $p_\theta(\mathbf{v}')$ from the network weights. For a sufficiently accurate conversion of hardware parameters to the parameters of the hardware distribution, this method can be applied to neuromorphic back-ends as well. For digital systems, this approach would be straightforward, whereas for analog ones such as BSS-2 it will need to rely on sufficient precision in the calibration data and in the analytical approximation of the sampled distribution.

Performance

In Figure 6 we compare a handcrafted, reasonably optimized C++ implementation of the Gibbs sampling algorithm with our spike-based implementation on BSS-2. Both methods are tasked to generate samples for multiple sizes of the physical system ($N \in \{4, 8, 12, 16, 20, 50, 100\}$) and multiple hidden layer sizes ($N_h \in \{20, 40, 80\}$).

To demonstrate the fundamental difference in scaling behavior we restrict the investigation to a single sampling run. In order to decrease the relative uncertainty of the measured timing we increase the number of samples to 10^6.

The CPU calculation is dominated by the number of synaptic interactions and thus runtime scales bilinearly with both the number of visible and number of hidden units, as can be seen in Figure 6. The CPU implementation ran in a single thread on a 2021 MacBookProM1Pro. Further improvement would be possible by e.g. using multiple Markov chains running in parallel on multiple cores.

For BSS-2 these systems fit on a single ASIC (for details see Methods A: Description of the BrainScaleS-2 ASIC) and therefore the runtime is network-size-independent. Consecutive samples can be taken every $\tau_{\mathrm{ref}}/2$ which, for our parametrization, results in the 5 s for generating 10^6 samples, shown in Figure 6.

A reduction of both $\tau_{\mathrm{syn}} = \tau_{\mathrm{ref}}$ should also be possible, which would further speed up the sampling generation, at the price of a more demanding calibration process and higher communication bandwidth requirements. This is achievable with modern manufacturing technologies, as predecessors of the BSS-2 architecture have already demonstrated higher speed-up factors of 10^5 and 10^4 w.r.t. biological real-time (Pfeil et al., 2013; Schemmel et al., 2010).