

ON-OFF neuromorphic ISING machines using Fowler-Nordheim annealers

Original

ON-OFF neuromorphic ISING machines using Fowler-Nordheim annealers / Xiao, Z., Akl, M., Leugring, J., Olajide, O., Malik, A., Dennler, N., Harper, C., Bose, S., Gonzalez, H.A., Samaali, M., Liu, G., Eshraghian, J., Pignari, R., Urgese, G., Andreou, A.G., Shankar, S., Mayr, C., Cauwenberghs, G., Chakrabartty, S.. - In: NATURE COMMUNICATIONS. - ISSN 2041-1723. - ELETTRONICO. - 16:(2025), pp. 1-13. [10.1038/s41467-025-58231-5]

Availability:

This version is available at: 11583/2999046 since: 2025-05-16T17:21:30Z

Publisher:

Springer Nature

Published

DOI:10.1038/s41467-025-58231-5

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

ON-OFF neuromorphic ISING machines using Fowler-Nordheim annealers

Received: 30 April 2024

Accepted: 17 March 2025

Published online: 31 March 2025

 Check for updates

Zihao Chen¹, Zhili Xiao¹, Mahmoud Akl², Johannes Leugring³, Omowuyi Olajide³, Adil Malik⁴, Nik Dennler^{5,6}, Chad Harper^{7,8}, Subhankar Bose¹, Hector A. Gonzalez^{2,9}, Mohamed Samaali², Gengting Liu², Jason Eshraghian¹⁰, Riccardo Pignari¹¹, Gianvito Urgese¹¹, Andreas G. Andreou¹², Sadasivan Shankar^{13,14}, Christian Mayr^{9,15}, Gert Cauwenberghs³ & Shantanu Chakrabartty¹ ✉

We introduce NeuroSA, a neuromorphic architecture specifically designed to ensure asymptotic convergence to the ground state of an Ising problem using a Fowler-Nordheim quantum mechanical tunneling based threshold-annealing process. The core component of NeuroSA consists of a pair of asynchronous ON-OFF neurons, which effectively map classical simulated annealing dynamics onto a network of integrate-and-fire neurons. The threshold of each ON-OFF neuron pair is adaptively adjusted by an FN annealer and the resulting spiking dynamics replicates the optimal escape mechanism and convergence of SA, particularly at low-temperatures. To validate the effectiveness of our neuromorphic Ising machine, we systematically solved benchmark combinatorial optimization problems such as MAX-CUT and Max Independent Set. Across multiple runs, NeuroSA consistently generates distribution of solutions that are concentrated around the state-of-the-art results (within 99%) or surpass the current state-of-the-art solutions for Max Independent Set benchmarks. Furthermore, NeuroSA is able to achieve these superior distributions without any graph-specific hyperparameter tuning. For practical illustration, we present results from an implementation of NeuroSA on the SpiNNaker2 platform, highlighting the feasibility of mapping our proposed architecture onto a standard neuromorphic accelerator platform.

Quadratic unconstrained binary optimization (QUBO) and Ising models are considered fundamental to solving many combinatorial optimization problems^{1–3} and in literature, both classical and quantum hardware accelerators have been proposed to efficiently solve QUBO/Ising problems^{4,5}. These accelerators use some form of annealing to guide the collective dynamics of the underlying optimization variables (e.g. spins) toward the global optima of the COP, which correspond to a specific system's ground energy states. Examples of such accelerators include superconducting qubit-based quantum annealers⁶, optical and digital coherent Ising machines (CIM) on^{7,8}, CMOS-based

oscillator networks^{9,10}, memristor-based Hopfield Network^{11–13}, and digital circuit-based simulated annealing (SA)^{14–17}. Quantum Ising machines that use quantum annealing can theoretically guarantee finding the optimal solution to the QUBO/Ising problem, however, the approach cannot yet be physically scaled to solve large-scale problems^{18–21}. On the other hand, classical QUBO/Ising solvers exploit nonlinear dynamics of the system to explore the solution space and avoid local minima. Simulated bifurcation machine (SBM)²² and probabilistic bit (p-bit)-based stochastic solvers relax the binary constraints and exploit the thermal noise inherent in magnetic tunnel

A full list of affiliations appears at the end of the paper. ✉ e-mail: shantanu@wustl.edu

junctions (MTJs) hardware^{23,24}. Memristor-based Hopfield Networks¹¹ also harness device-intrinsic noise and can achieve low energy-to-solution and time-to-solution metrics. However, if the objective is to produce solutions that consistently approach the state-of-the-art (SOTA) optimization metric or even exceed the SOTA (discover better solutions), then the precision of computation and the dynamic range of peripheral read-out circuits limit the performance of analog Ising machines. This is highlighted in Fig. 1a for three classes of COPs: low-complexity (L), medium-complexity (M), and high-complexity (H). Here, the complexity of the problems is characterized based on various metrics such as number of variables, graph density, and graph transitivity. For low-complexity problems, most solvers can find the

SOTA (or the ground state) for every run. This results in a distribution that is concentrated around SOTA, as highlighted in Fig. 1a. However, for COPs with higher complexity, the distribution of solutions can exhibit a large variance, because: (a) the ground state could be significantly separated from the SOTA (shown in Fig. 1a) and (b) even finding an approximate solution could be NP-hard (or requires exponential time or high precision). Scaling analog Ising machines architecturally to larger problems would also require tiling small-sized memristor/analog crossbars with additional communication infrastructure (using spikes or digital encoding). Digital Ising architectures do not suffer from the precision and scaling challenges. Furthermore, in memristor-based Ising machines, the statistics of the device noise

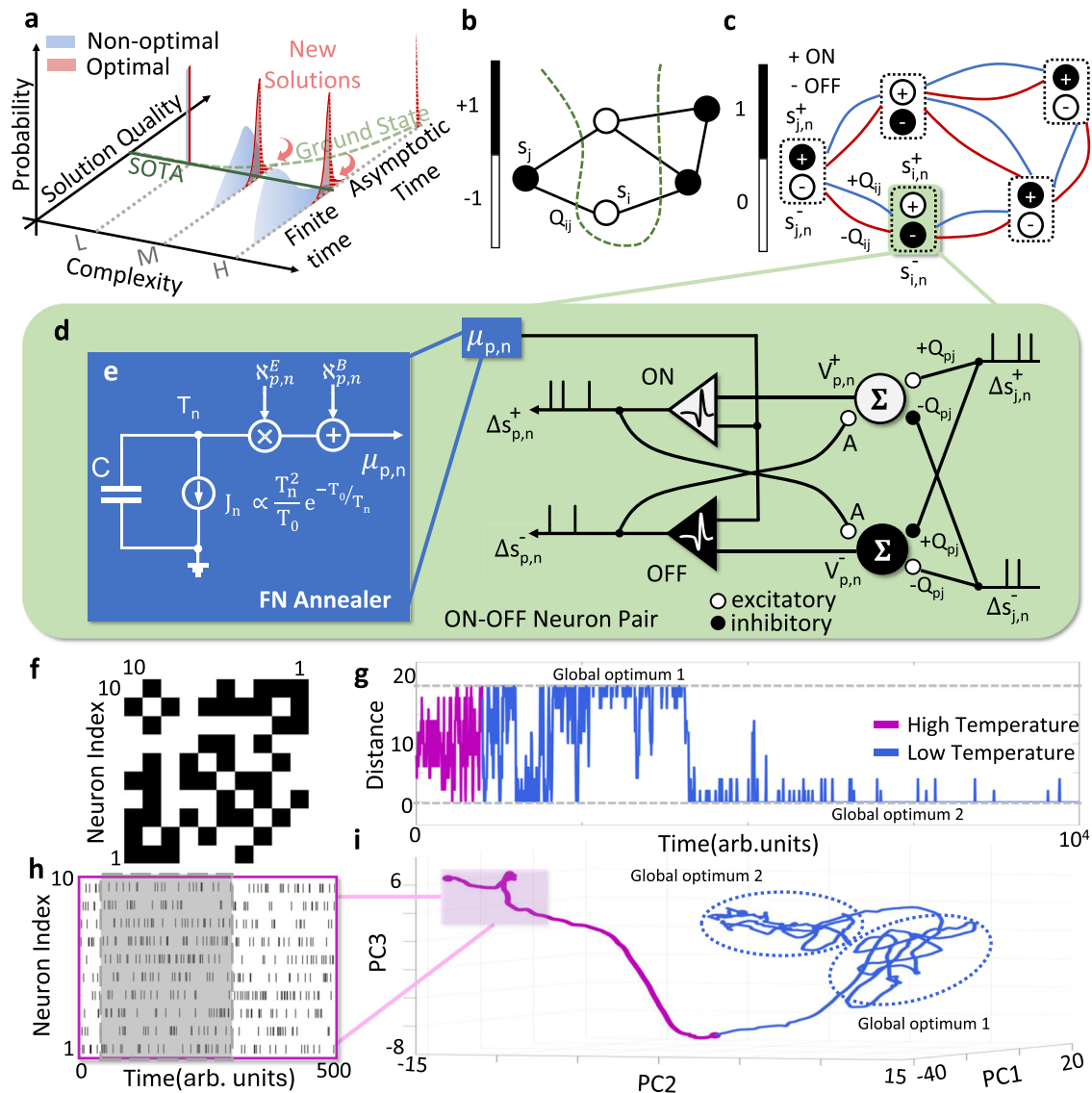


Fig. 1 | NeuroSA motivation for mapping of optimal simulated annealing into a neuromorphic architecture. **a** Illustration of the distribution of solutions generated by optimal and non-optimal Ising machines for different COP complexity: Low (L), Medium (M), and High (H). An ideal Ising/QUBO solver produces distribution of solutions that is concentrated near the SOTA and has the potential to produce novel, previously unknown solution that is closer to the Ising ground state; A MAX-CUT problem defined over a **b** graph with weights Q_{ij} which is decomposed into **c** pairs of ON-OFF neurons by NeuroSA. **d** Each ON-OFF integrate-and-fire neurons are coupled to each other by an excitatory synapse with weight A and the pair is connected differentially to other ON-OFF neuron pairs through the synaptic weights $Q_{ij}, -Q_{ij}$. The thresholds for both ON-OFF neurons are dynamically

adjusted by an **e** FN annealer which comprises an FN integrator, an exponentially distributed noise source \mathcal{N}_n^E and a Bernoulli noise source \mathcal{N}_n^B . Illustration of NeuroSA dynamics for a MAX-CUT graph with 10 vertices connected by a weight matrix Q shown in **(f)**. **g** Evolution of the distance between the solutions generated by NeuroSA to the two known ground state solutions at a given time-instant which highlights the escape mechanisms in the high- and low-temperature regimes. **h** Raster plot of aggregated spiking activity generated by the ON and OFF neuron pairs, and **i** visualization of the NeuroSA trapping and escape dynamics using a principal component analysis (PCA)-based projection of the network spiking activity estimated within a moving time-window.

and its modulation have to be experimentally tuned/calibrated such that the results approach the state-of-the-art (SOTA) solution quality. There is no theoretical guarantee that such a machine would improve upon the SOTA or asymptotically converge to the ground state.

Advances in neuromorphic hardware have now reached a point where the platform can simulate networks comprising billions of spiking neurons and trillions of synapses. Implementations of these neuromorphic supercomputers range from commercial-off-the-shelf CPU-, and GPU-based platforms^{25,26} to custom FPGA-, multi-core-, and ASIC-based platforms^{27–29}. As an example, the SpiNNaker2 microchip³⁰, which has been used for some experiments in this paper, can integrate more than 152,000 programmable neurons with more than 152 million synapses in total. Although the primary motivation for developing these platforms has been to emulate/study neurobiological functions^{31,32} and to implement artificial intelligence (AI) tasks^{33–35} with much needed energy efficiency³⁶, it has been argued that the neuromorphic advantage can be demonstrated for tasks that can exploit noise and nonlinear dynamics inherent in current neuromorphic systems. These approaches exploit the emergent properties of an energy or entropy minimization process^{37,38}, phase transitions and criticality³⁹, chaos⁴⁰, and stochasticity^{41–43}, both at the level of an individual neurons^{22,44,45} and at the system level such as the Hopfield Network^{11,37} and Boltzmann machines^{46,47}. Recently, neuromorphic advantages in energy efficiency have been demonstrated for solving optimization problems^{34,48} and for simulating stochastic systems like random walks⁴⁹. These specific implementations exploit the high degree of parallelism inherent in neuromorphic architectures for efficient Monte Carlo sampling, and for implementing Markov processes both of which are important for solving Ising problems. Other promising approach has been to use neuromorphic architectures to solve semi-definite programming (SDP), which is an approximation method to solve many COPs⁵⁰. In fact, under the correctness assumption of the unique games conjecture⁵¹, SDPs have been employed to produce optimal solutions amongst all polynomial-time algorithms⁵⁰. Examples of such implementations include the mapping of the Goemans–Williams (GW) SDP onto Intel Loihi^{52,53} for MAX-CUT COPs. However, GW and other SDP algorithms only provide lower bounds on the solution quality and not on the variance of distribution of solutions. Furthermore, the precision required to solve the SDP algorithm could be high to be able to achieve the SOTA. As a result, even achieving the SDP approximation quality bound precisely or exceeding the bound (for specific graphs and runs) still requires a brute-force (exponential or sub-exponential) search around the SOTA. Also note that for COPs like MAX-CUT, even an improvement by a single cut could represent a significant step, as the improvement could only be achieved by finding a novel solution.

Simulated Annealing (SA) algorithms, on the other hand, can provide guarantees of finding the QUBO/Ising ground state, provided the annealing schedule follows specific dynamics^{54–56}. Previous attempts to solve Ising problems using neuromorphic hardware^{12,13} emulated SA dynamics, which did not guarantee asymptotic convergence to the Ising ground state. Hence, a neuromorphic architecture that is functionally isomorphic to the SA algorithm with an optimal annealing schedule should produce high-quality solutions across different runs. This feature is highlighted in Fig. 1a by the desired (or optimal) distribution that is concentrated near the SOTA. Furthermore, as shown in Fig. 1a, having an asymptotic optimality guarantee will also ensure that a long run-time might produce a solution that is at least better than the current SOTA, if the current SOTA is not already the ground state of the COP. However, the dynamics of the SA algorithm can be very slow which motivates its mapping onto a neuromorphic architecture and accelerator.

How can optimal simulated annealing algorithms be mapped onto large-scale neuromorphic architectures? The key underpinnings of any neuromorphic architecture are: (a) asynchronous (or Poisson)

dynamics that are generated by a network of spiking neurons; and (b) efficient and parallel routing of spikes/events between neurons across large networks. Both these features are essential for solving the Ising problem and efficient mapping of SA onto neuromorphic architecture. In its general form Ising problem minimizes a function (or a Hamiltonian) $H(\mathbf{s})$ of the spin state vector \mathbf{s} according to

$$\min_{\mathbf{s} \in \{-1, +1\}^D} H(\mathbf{s}) = \frac{1}{2} \mathbf{s}^T \mathbf{Q} \mathbf{s} + \mathbf{b}^T \mathbf{s} \quad (1)$$

where $\mathbf{b} \in \mathbb{R}^D$ represents an external field or bias vector and can also be used to introduce additional constraints into the Ising problem⁵⁷. In SI Sections S1, 2, and 3 we consider these specific cases, but for the ease of exposition, our focus will be on problems where $\mathbf{b} = \mathbf{0}$. In such cases the Ising problem becomes equivalent to the MAX-CUT problem (shown in SI Sections S1 and S2), which is easy to visualize and analyze⁵⁸. For a simple MAX-CUT graph depicted in Fig. 1b, each of the spin variables, denoted as $s_i \in \{-1, +1\}$, where $i = 1, \dots, D$, is associated with one of the D vertices in the graph \mathcal{G} . The graph's edges are represented by a matrix $\mathbf{Q} \in \mathbb{R}^{D \times D}$, wherein Q_{ij} signifies the weight associated with the edge connecting vertices i and j . Given the graph \mathcal{G} , the objective of the MAX-CUT problem is to partition the vertices into two classes, maximizing the number of edges between them. If an ideal asynchronous operation is assumed (see “Methods” subsection “Asynchronous Ising machine model”), then at any time instant n , only one spin (say the p th spin) changes its state by $\Delta s_{p,n} \in \{-1, 0, +1\}$. In this case, the function H decreases or $\Delta H_n < 0$, if and only if the condition

$$\Delta s_{p,n} \left[\sum_{j=1}^D Q_{pj} s_{j,n-1} \right] < 0 \quad (2)$$

is satisfied. The inherent parallelism of neuromorphic hardware ensures that the pseudo-gradient $\sum_{j=1}^D Q_{pj} s_{j,n-1}$ is computed at a rate faster than the rate at which events $\Delta s_{p,n}$ are generated. The condition described in Eq. (2), when combined with the simulated annealing's probabilistic acceptance criterion⁵⁴, leads to a neuromorphic mapping based on coupled ON–OFF integrate-and-fire neurons where the p th ON–OFF pair is shown in Fig. 1d. Please refer to the Methods section 4 for the derivation of the ON–OFF neuron model. The p th ON–OFF neuron pair is differentially connected to the j th neuron pair through the synaptic weights Q_{pj} , $-Q_{pj}$. The spikes generated by this p th post-synaptic ON–OFF neuron pair $\Delta s_{p,n}^+$, $\Delta s_{p,n}^- \in \{0, 1\}$ differentially encode the change in the p th spin state, and the cumulative state $s_{j,n}$ of the pre-synaptic neuron is estimated by continuously integrating the input spikes $\Delta s_{j,n}^+$, $\Delta s_{j,n}^-$ received from the j th neuron. To ensure that the spiking activity of the ON–OFF neuronal network is functionally isomorphic to the acceptance/rejection dynamics of an SA algorithm, the firing threshold $\mu_{p,n}$ of the p th neuron adjusted over time by a Fowler–Nordheim (FN) annealer, shown in Fig. 1d.

We employ a Fowler–Nordheim dynamical system that can produce dynamic thresholds that correspond to the SA optimal annealing schedule. One of the key results from the SA literature^{55,56} is the proposition that a temperature cooling schedule that follows $\sim \frac{c}{\log(1+n)}$ can guarantee asymptotic convergence to the QUBO/Ising ground state, where c denotes the largest depth of any local minimum of Ising Hamiltonian $H(\mathbf{s})$ and n denotes the discrete time step. A dynamical systems model in Fig. 1e comprising of a time-varying Fowler–Nordheim (FN) current element J_n ⁵⁹ can generate the optimal $T_n = \frac{T_0}{\log(1+n/C)}$ according to⁶⁰, where T_0 and C are Fowler–Nordheim annealing hyperparameters (see the “Methods” subsection “Dynamical systems model implementing the FN Annealer”). The FN dynamics can either be generated using a physical FN-tunneling device⁶⁰ or can be emulated by implementing an FN-tunneling dynamical systems model on digital hardware. Here, we chose the digital emulation because of

the precision required for SA in the low-temperature regime. The FN dynamics can then be combined with the independent identically distributed (i.i.d.) random variables $\mathcal{N}_{p,n}^E$ and $\mathcal{N}_{p,n}^B$ to determine the dynamic firing threshold $\mu_{n,p}$ for each ON–OFF integrate-and-fire neuron pair p . $\mathcal{N}_{p,n}^E$ is drawn from an exponential distribution whereas $\mathcal{N}_{p,n}^B$ is drawn from a Bernoulli distribution with values $\{0, 1\}$. The choice of the two distributions ensures that every neuron has a finite probability of firing, which is equivalent to satisfying the irreducibility and aperiodicity conditions in SA.

The ON–OFF neuron pair and the integrated FN annealer, shown in Fig. 1c form the basic computational unit of NeuroSA which can be used to solve Ising problems on different neuromorphic hardware platforms. Due to the functional isomorphism between NeuroSA and the optimal SA algorithm, the hardware could accelerate and asymptotically approach the Ising ground state, as highlighted in Fig. 1a. We show in the results that even when NeuroSA solves a COP for a finite duration, the machine produces the distribution of solutions that are concentrated around the SOTA, as shown in Fig. 1a, and this is achieved without significant tuning of hyperparameters.

Results

We first demonstrate the dynamics of the NeuroSA architecture for a small MAX-CUT problem, where the optimization landscape and the search dynamics can be easily visualized and where the Ising ground states can be determined using brute-force search. We then evaluate NeuroSA on larger-scale benchmark COPs for which the SOTA solution quality is well documented in the literature^{61,62}.

Experiments using small-scale graphs

NeuroSA is first applied to a 10-node MAX-CUT graph described by the interconnection matrix \mathbf{Q} shown in Fig. 1f. For this graph, the two degenerate ground states exist (due to the gauge symmetry where $\mathbf{s} \leftrightarrow -\mathbf{s}$) and can be found by an exhaustive search over all possible spin state configurations. Fig. 1g plots the Manhattan distance measured with respect to one of the ground states. For this specific example, there exist two ground states (due to gauge symmetry) which are labeled as global optimum 1 and 2 in Fig. 1g. It is evident from Fig. 1g that, like SA, the dynamics of NeuroSA can be categorized into two phases: the high-temperature regime and the low-temperature regime. These regimes are determined by the firing threshold $\mu(t)$ in Fig. 1d. To visualize the network dynamics, the aggregated spiking rate for each of the ON–OFF neurons is calculated using a moving window as shown in Fig. 1h. The population firing dynamics are projected onto a reduced 3-D space spanned by its three most dominant principal component vectors, as shown in Fig. 1i. This principal component analysis (PCA)-based approach is a standard practice in analyzing spiking data⁶³ and details of the approach are described in the “Methods” subsection “Generation of network PCA trajectories”. In the high-temperature phase, the network dynamics evolve along the network gradient, resulting in faster convergence along a smooth trajectory as depicted in Fig. 1i. As the temperature cools down, NeuroSA exploration is trapped in the neighborhood of one of the two global optima, as depicted in Fig. 1g, i. In the neighborhood, the dynamics follow a random walk; however, the dynamics can periodically escape one of the global optima for further exploration and possibly converge to the second global optimum. This is highlighted in Fig. 1i by the continuous trajectory connecting the two optima/attractors. Eventually, as shown in Fig. 1g, due to the annealing process the dwell-time of dynamics in the neighborhood of the optima increases over time. Note that state-space exploration in the low-temperature regime is a significant problem in SA algorithms, and heuristics such as hybrid quantum-classical methods⁶⁴ have been proposed to accelerate this process. In NeuroSA, the Fowler–Nordheim dynamical process allows for a finite probability of escape even in the low-temperature regime.

Experiments using medium-scale graph

We next apply NeuroSA to a MAX-CUT problem on a graph where the ground state is not known, but the SOTA solution is well documented. We chose the G15, an 800-node, binary weighted, planar graph⁶⁵ for which the SOTA is 3050 cuts²². The NeuroSA architecture was simulated on a CPU-based platform and the hardware mapping procedure is described in the “Methods” subsection “Estimation of empirical probability density function” and the pseudo-code for the implementation is presented in SI Section S4. Figure 2a shows the solution found by NeuroSA over time converging to the SOTA states.

The dynamics of the noisy firing threshold μ_n is shown in Fig. 2b and is bounded by $T_n \sim \frac{1}{\log(n)}$ (depicted in the figure as the dotted line), produced by the FN annealer. As the envelope of the threshold decreases, it inhibits the firing probability of neurons as shown by the histogram in Fig. 2c. During the initial phases of convergence, there is a gap between the $\frac{1}{\log(n)}$ envelope and the number of active neurons (or the neurons whose membrane potential exceeds the firing threshold). Thus, in this phase, the dynamics of the network seem to be governed by the network gradient. However, the tails of the histogram fit the $\frac{1}{\log(n)}$ dynamics reasonably well, highlighting the influence of the FN-based escape mechanism on the network dynamics. The network population dynamics are depicted by the PCA trajectory of the aggregated spiking rate and are plotted in Fig. 2d. Similar to the results for the small-scale graph in Fig. 1i, the trajectory reflects the evolution of the NeuroSA system as it explores the solution space. The exploration in the high-temperature regime follows a more confined trajectory in the PCA space, indicating that the network dynamics evolve in the direction of the Ising energy gradient. As the temperature cools down near convergence, the NeuroSA dynamics are dominated by the random walk and sporadic escape mechanisms with no specific direction, as shown in Fig. 2d.

Figure 2e plots the distribution of the solution quality (normalized with respect to SOTA) when different cooling schedules are chosen and the r.v. N^E are chosen from different statistical distributions. In particular, previous neuromorphic Ising models and stochastic models have used Gaussian noise as a mechanism for asymptotic exploration and for escaping local minima. However, the results in Fig. 2e show that this approach produces distributions with longer tails and in some cases solutions that are significantly worse than the SOTA. Only for an FN annealer and an exponentially distributed noise N^E , the distribution of solutions obtained by NeuroSA is more concentrated around the SOTA.

Benchmarking NeuroSA for different MAX-CUT graphs

Next, the NeuroSA architecture was benchmarked for solving MAX-CUT problems on different Gset graphs. Figure 3 provides a detailed evaluation of the NeuroSA algorithm’s performance on the Gset benchmarks, with results generated using both traditional CPU-based and the SpiNNaker2 platform. The architecture is configured similarly for both hardware platforms and across all benchmark tests. This uniformity is important, as it demonstrates that NeuroSA’s performance is robust across and agnostic to different MAX-CUT graph complexities. Also, it obviates painstaking hyperparameter tuning for each set of graphs or problems.

Figure 3a–d reports the distribution of solutions obtained by NeuroSA for different Gset MAX-CUT benchmarks. For each NeuroSA run, the solution (number of cuts) is normalized with respect to the SOTA solution which is highlighted in Fig. 3a–d as the unity marker. Note that for MAX-CUT problems, the SOTA is from previously reported results in literature^{22,61}. The probability density function is empirically obtained as described in the “Methods” subsection “Estimation of empirical probability density function”. While determining the complexity of COPs is in itself a difficult task in itself, to gain more insight, the distributions are organized based on commonly used graph complexity metrics. Figure 3a–d organizes the results by: (a) the

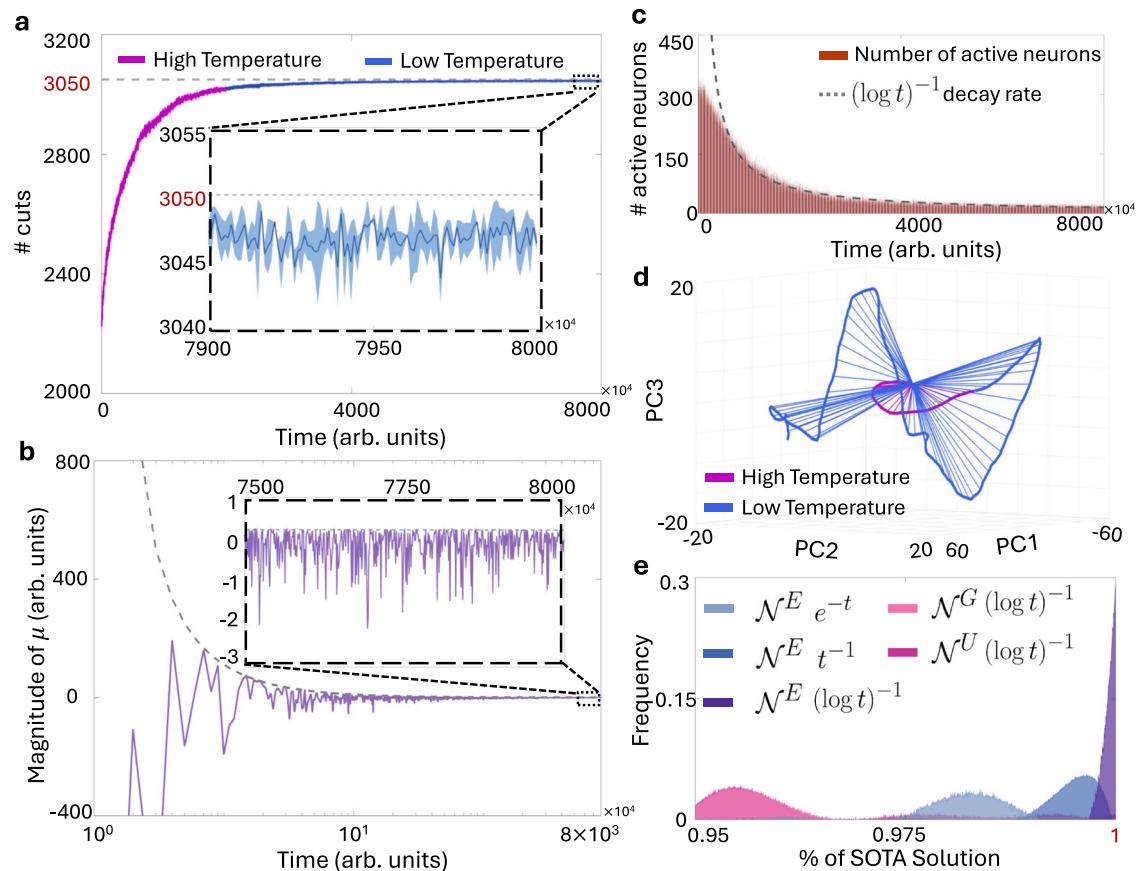


Fig. 2 | NeuroSA dynamics for the G15 MAX-CUT graph with 800 vertices and 4661 edges. **a** Convergence plot showing steady increase in the solution quality with the inset showing fluctuations near 3050 cuts which is the current SOTA for this graph. **b** Dynamics of the firing threshold with inset showing sparse but large fluctuations that trigger escape mechanisms. **c** Plot showing the number of active neurons decaying following $\sim \frac{1}{\log t}$ without the contribution of the Bernoulli r.v. \mathcal{N}^B .

d PCA trajectory of the NeuroSA dynamics where the initial (high temperature) regime follows a path defined by the network gradient and the trajectory near convergence (or low-temperature path) exhibits expanding exploration of the solution space. **e** Distribution of the G15 solutions obtained for different annealing schedules (e^{-t} , $(\log t)^{-1}$, t^{-1}) and noise statistics (exponential—denoted by \mathcal{N}^E , Gaussian—denoted by \mathcal{N}^G , and Uniform—denoted by \mathcal{N}^U).

graph size, namely the number of vertices; (b) the average fan-out per node; (c) the graph entropy, measuring the randomness in connectivity⁶⁶; and (d) the network transitivity, focusing on node connectivity density, indicating clustering within the network⁶⁷. The average fan-out per node measures the typical number of direct connections (outgoing edges) each node has, providing a basic indication of the graph's overall connectivity and potential for information spread. Graph entropy, on the other hand, quantifies the randomness or disorder in the graph by analyzing the distribution of these connections across all nodes. It offers insights into how evenly or unevenly the connections are distributed, with higher entropy indicating a more complex or disordered network structure. The global clustering coefficient, also known as transitivity, measures the degree to which nodes in a graph tend to cluster together. This coefficient assesses the overall tendency of nodes to create tightly knit groups, with higher values suggesting a greater prevalence of interconnected triples of nodes, which can indicate a robust local structure within the network. While the average fan-out per node provides a simple measure of connectivity, it does not capture the nuances of how these connections are configured, which is where graph entropy and the global clustering coefficient come into play. Graph entropy complements the average fan-out by assessing the variability in node connectivity, highlighting potential inequalities or irregularities in how nodes are linked. In contrast, the global clustering coefficient focuses on the tendency to form local groups, offering a view of the graph's compactness and the likelihood of forming tightly connected

communities. Together, these metrics provide a multi-dimensional view of a graph's complexity, indicating not only how many connections exist, but also how they are organized and how they foster community structure and network resilience.

The results show that the NeuroSA solutions consistently reach within 99% of the SOTA, despite of the complexity of the graph. However, as expected, the variance of the distribution does increase with complexity which indicates that one can also speculate on the general complexity of the COP itself. In Fig. 3e we show that there exists an algorithmic advantage in mapping NeuroSA onto a parallel architecture, like the SpiNNaker2 platform. We compare the results obtained by five NeuroSA instances launched in parallel, where each instance executes for 10^8 iterations against a singleton NeuroSA instance that runs for 5×10^8 iterations. We then aggregate the results from the five parallel instances by taking the average, and plot the probability distribution of the quality of solution for both parallel and singleton instances. As shown in Fig. 3e, the parallel search results in a more concentrated distribution around the SOTA solution than the long-running singleton approach. Effectively, the parallel NeuroSA instances take different trajectories to the solution and the aggregation of multiple searches results in amplifying the probability of reaching the neighborhood of the SOTA/ground state solution within a finite time constraint. However, we would like to point out that the optimal annealing schedule promises global optimality only in the asymptotic (exponential) time limit. Therefore, improvements on the current solution to reach the ground state using a parallel approach

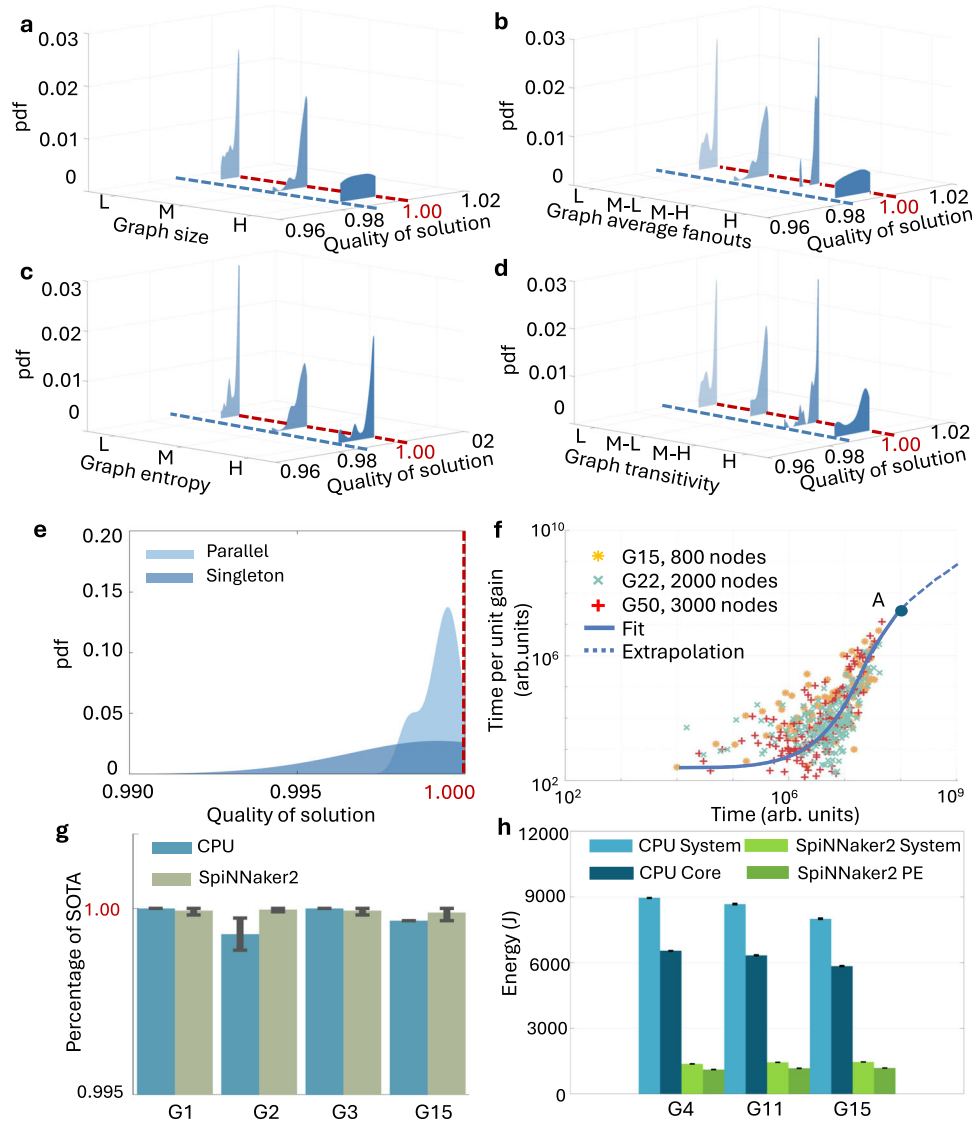


Fig. 3 | NeuroSA results for MAX-CUT and MIS benchmarks. **a–d** Empirical probability density functions (pdf) of the solutions on Gset benchmarks⁶¹. The solutions obtained after 10^8 iterations fall within the interval (0.989, 1.00) of SOTA, indicated by the red and blue dotted lines. The results are ordered with increasing complexity: Low (*L*), Medium-Low (*M-L*), Medium (*M*), Medium-High (*M-H*), High (*H*). For complexity metric: (a) the number of graph vertices, $L = \{800, 1000\}$, $M = \{2000, 3000\}$, and $H = \{5000, 10,000\}$. **b** the average fan-out, where $L = 2.0$, $M-L = 6.0$, $M-H = 10.0$, $H = 24.0$. **c** The graph entropy, where $L = [0, 2)$, $M = (2, 4)$ and

$H = (4, 5)$. **d** The graph transitivity, where $L = [0, 0.001]$, $M-L = (0.001, 0.05]$, $M-H = (0.05, 0.14)$, $H = [0.14, 0.16]$. **e** Parallel search comprising of 5 NeuroSA instances yields more consistent results than a single search running for 5 times the duration of the parallel search. **f** Instantaneous time per unit gain in solution for three different Gset benchmarks approaches an exponential/sub-exponential run-time. **g** Results from NeuroSA implemented on the SpiNNaker2 neuromorphic platform. **h** The energy-to-solution results on both CPU and SpiNNaker2 platforms.

would still rely on long run-time of one of many singleton NeuroSA instances. We extended the execution time by 1000 \times , to 10^{11} iterations to see if NeuroSA could find a novel solution for a few sets of MAX-CUT problems. Unfortunately, within finite run-time (both on CPU and SpiNNaker2), NeuroSA was unable to discover solutions that exceeded the SOTA for the Gset/MAX-CUT benchmarks.

To understand the difficulty in finding new solutions, Fig. 3f presents an analysis of the time required per unit gain in the solution for three distinct Gset benchmarks. The plots reveal an increasing cost (computational time) to achieve marginal improvement in the quality of the solution. The key metric in the plot Fig. 3f is the ratio between the time needed to obtain a unit improvement to the total run time. As it becomes harder to find better solutions, the ratio tends to unity (run-time becomes exponential or sub-exponential) This is shown by the extrapolation curve and the transition point *A*, which might highlight the point of diminishing returns. This could be

taken as a hardware-agnostic stopping criterion (polynomial run-time) for a COP.

Figure 3g shows the results when NeuroSA is implemented on the SpiNNaker2 platform (details provided in SI Section S5) for some of the Gset benchmarks. The results show similar or better solutions than the CPU/software implementation of NeuroSA. Note that NeuroSA is a randomized algorithm so each hardware implementation will exhibit different variance based on the quality of noise generator in the FN-annealer. The performance advantage achieved by mapping NeuroSA onto SpiNNaker2 is summarized in SI Section S5.2 and in Fig. 3h which show the energy-to-solution comparison with respect to the CPU-based implementation. The comparison is shown for three different 800-node MAX-CUT problem, and we compare the energy dissipated by the system (board-level) and by the processing-element-level (PE). Due to the distributed and multi-core nature of the SpiNNaker2 system, the PE energy also accounts for the inter-core

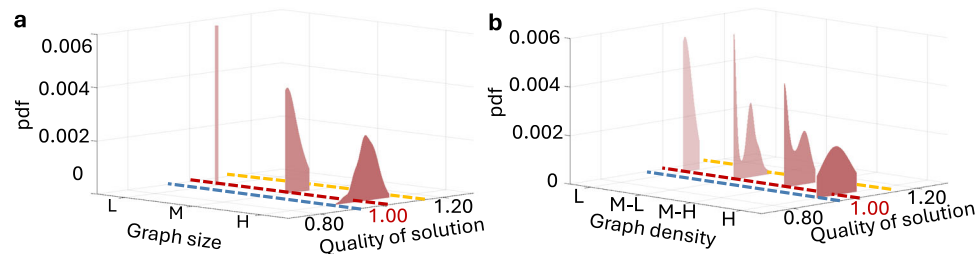


Fig. 4 | NeuroSA results for MIS benchmarks showing that all solutions fall within the interval (0.944, 1.11) of the SOTA (marked by dotted lines). Solutions are ordered according to complexity metrics: **a a number of graph vertices, with**

L = {10, 25, 50}, **M** = {100, 250, 500} and **H** = {1000, 2500, 5000}, and **b** graph density, with **L** = 0.01, **M-L** = 0.05, **M-H** = 0.1, **H** = 0.25.

communication consumption over the NoC. As shown in Fig. 3h, in either case, the SpiNNaker2 implementation outperforms the CPU-based implementation in terms of energy-to-solution for the same workload. Furthermore, this performance advantage is evident even if the current SpiNNaker2 implementation of NeuroSA is not fully optimized.

Benchmarking NeuroSA for MIS problems

For the next set of experiments, the performance of NeuroSA architecture was evaluated for solving the maximum-independent-set (MIS) COPs. The problem involves searching for the largest set of vertices in a graph such that no two vertices in that set share an edge. Since the solutions to the MIS problem are subject to constraints, the problem is inherently harder than MAX-CUT where every state configuration is considered a valid solution. Details of mapping MIS COP onto an Ising formulation which can then be solved using NeuroSA are provided in SI Section S3. We evaluated NeuroSA on various MIS graphs from the NeuroBench benchmark suite⁶² and we characterized the distribution of the solutions generated for different graph sizes and graph densities, as shown in Fig. 4. Graph density represents the density of connections among vertices which controls the dimensionality of the system's state space and the effective dimensionality of the Ising energy landscape. Furthermore, higher density also alters the eigenvalue spectrum of the Hamiltonian, broadening it, and creating a richer set of critical points on the Ising energy surface. Similar to the MAX-CUT results, the MIS results also produce distributions are concentrated, without any hyperparameter tuning. However, unlike the MAX-CUT benchmarks, NeuroSA consistently finds solutions that are better than the current SOTA for MIS benchmarks. Also, note that the distributions in Fig. 4 are bounded below by 95% of the SOTA, and this deviation occurs when the optimal set size just changes by 1. Even though the variance of the distribution becomes larger with increasing complexity, a significant portion of the solutions surpasses the current SOTA, which validates the objective of the NeuroSA in Fig. 1a.

Discussion

In this paper, we proposed a neuromorphic architecture called NeuroSA that is functionally isomorphic to a simulated annealing optimization engine. The isomorphism allows mapping optimal SA algorithms to neuromorphic architectures, providing theoretical guarantees of asymptotic convergence to the Ising ground state. The core computational element of NeuroSA is formed by an ON-OFF integrate-and-fire neuron pair that can be implemented on any standard neuromorphic hardware. Hence, NeuroSA can exploit the computational power of both existing and upcoming large-scale neuromorphic platforms, such as SpiNNaker2 and HiAER-Spike. Inside each ON-OFF neuron pair is an annealer whose stochastic properties are dictated by a Fowler-Nordheim (FN) dynamical system. Collectively, the neuron model and the FN annealer generate population activity that emulates the sequential acceptance and rejection dynamics of the SA algorithm.

The functional isomorphism between NeuroSA and the optimal SA algorithm also enables us to draw insights from SA dynamics to understand the emergent neurodynamics of NeuroSA and its convergence properties to a steady-state solution. For instance, the Bernoulli r.v. \mathcal{N}_n^B within the FN annealer ensures the asynchronous firing such that only one of the neurons in NeuroSA fires at any given moment. From the perspective of SA, this asynchronous decomposition ensures that each combinatorial step of the COP is tractable, as described by the mathematical condition in Eq. (9). Also, the use of i.i.d Bernoulli r.v.s in each ON-OFF neuron pair ensures that any pair can potentially fire (if its firing criterion is met), which in turn ensures that NeuroSA satisfies a key ergodic convergence criterion similar to that of SA algorithms. According to this criterion, every potential Ising state is reachable⁵⁴. The exponentially distributed r.v. \mathcal{N}_n^E in the FN annealer upholds that an equivalent detailed balance criterion in SA⁵⁴ is satisfied, thereby ensuring that the NeuroSA network attains an asymptotic steady-state firing pattern. This steady-state pattern corresponds to different mechanisms of exploring the Ising energy states, as depicted by the PCA network trajectories in Figs. 1i and 2d, with the assumption that the exploration will asymptotically terminate near the Ising ground state. This asymptotic convergence is guaranteed by modulating the dynamic threshold μ_n in the ON-OFF neuron pair, mimicking the optimal $\mathcal{O}(1/\log)$ temperature schedule in SA proposed by^{55,56}. Any choice of distribution other than the exponential distribution for the r.v. \mathcal{N}_n^E will violate the SA's detailed balance criterion and hence the network might not encode a steady-state distribution.

In this work, two families of COPs, MAX-CUT and MIS, have been selected as COP benchmarks because they are well-studied COPs and the SOTA results for different MAX-CUT graphs are well documented in literature⁶⁸. As shown in the “Results” subsection “Benchmarking NeuroSA for different MAX-CUT graphs”, the NeuroSA architecture can consistently find solutions that are closer than 99% SOTA metrics for different MAX-CUT benchmarks and for the recently proposed MIS benchmark suite, NeuroSA improves on the current SOTA. Note that the ground state solution for most of these graphs is still not known. In this regard, the asymptotic convergence to the ground state offered by the NeuroSA architecture is important as it can ensure good quality solutions across different runs, as highlighted in Fig. 3a–d. It is important to note that the $\mathcal{O}(1/\log)$ annealing schedule could make the convergence significantly slow which can be mitigated by sheer hardware acceleration offered by current and next-generation neuromorphic platforms. Also note that in Fig. 3a–d, the solutions obtained for some MAX-CUT graphs are inferior (percentage relative to the SOTA) compared to others, irrespective of the problem size (number of spin variables or ON-OFF neurons). This is because the problem complexity of some of the MAX-CUT problems is higher which implies that the NeuroSA architecture has to explore different regions of the energy landscape. By increasing the simulation run time and choosing a larger value of the hyperparameter T_0 , the quality of the solution can be improved for all MAX-CUT graphs. The NeuroSA Ising machine can be used to solve other COP-like Hamiltonian path problems or Boolean

satisfiability problems as well by optimizing a similar form of $H(\mathbf{s})$ in Eq. (3) but with real-valued Q_{ij} . However, the energy landscape of the resulting $H(\mathbf{s})$ is more complex and hence would require different choices of T_0 and the simulation time to achieve SOTA solutions.

The NeuroSA architecture relies on the asynchronous nature of the SA acceptance dynamics which is directly encoded by spikes. The underlying assumption is that the spike from the neuron is propagated to all its synaptic neighbors before any other neuron in the network spikes. As shown by Eqs. (15) and (16), spike propagation from the ON–OFF neuron pair is equivalent to propagating pseudo-gradients in an SA algorithm. Most large-scale neuromorphic platforms rely on event routing mechanisms like Address event routing to transmit spikes across the network which incurs latency. As a result, if the spiking rate (equivalently the rate of the number of acceptances) is high, the asynchronous criterion specified in Eq. (9) might not be satisfied. Furthermore, in practice, spikes (or event packets) might not be properly routed to the neurons or dropped. Due to the stochastic nature of the NeuroSA algorithm, these artifacts or errors can be tolerated during the initial phases of the convergence process. Asymptotically, as the network spiking rates decrease or the inter-spike interval increases, there would be enough time between events for the pseudo-gradient information to be correctly routed and hence Eq. (9) is satisfied. This region of convergence corresponds to the low-temperature regime where it is important to explore distant states and at the same time accept proposals (or produce spikes) only when the network energy decreases.

The tolerance of the NeuroSA architecture to communication errors provides a mechanism to accelerate its convergence using a hybrid approach as mentioned previously. In the initial phase of hybrid approach, the optimization endeavor follows the steepest gradient descent and performs parallel updates on neuron states resembling SDP. The expected value of a solution found in the initial phase can be characterized by the GW-like bounds⁵⁸. Then, the search follows the brute-force approach where NeuroSA can asymptotically approach the ground state and in the process exploring novel solutions that are better than the SOTA. As shown in SI Fig. S5, when NeuroSA is initialized at a low temperature (less noisy threshold) initially, the architecture converges to a solution that is >95% of SOTA. The convergence in this case is 10^4 times faster than the case when NeuroSA is initialized at a warmer temperature (more noisy threshold). To avoid getting trapped in the neighborhood of a local attractor, the system temperature is increased and annealed according to the optimal cooling schedule, as depicted in SI Fig. S5. Note that the asymptotic performance is still determined by the $\mathcal{O}(1/\log)$ decay and the cold–warm acceleration does not improve the quality of the solution. Furthermore, as highlighted in Fig. 3f, as the optimization proceeds, the time needed to achieve a unit gain in the quality of the solution increases with time, with the last gain consuming the majority of the entire simulation duration. Consequently, accelerating NeuroSA's initial convergence using a low-temperature start might not significantly reduce the overall time-to-solution when the goal is to approach the asymptotic ground state. However, the approach does enhance the efficiency of the NeuroSA to approach SOTA solutions under real-time constraints.

One of the attractive features of the NeuroSA mapping is that the architecture can be readily implemented and scaled up on existing neuromorphic platforms like SpiNNaker2, especially given the availability of large-scale systems such as the 5-million cores supercomputer in Dresden³⁰ with more than 35K SpiNNaker2 chips interconnected in a single system. The algorithmic advantage of mapping NeuroSA on a parallel architecture was demonstrated in the small-scale experiment shown in Fig. 3e. Effectively, a parallel search amplifies the chance of approaching the SOTA/ground state within finite time-interval, because of the inherent independence of the brute-force search in the low-temperature regime. This implies that multiple copies of NeuroSA can be instantiated simultaneously on SpiNNaker2 without incurring little to no

overhead, as long as local memory overflow is avoided. As the problem scales, memory mapping and encoding in neuromorphic architectures like SpiNNaker2 becomes the main challenge and bottleneck. However, the inherent sparsity in the COPs (defined by the matrix \mathbf{Q} can be leveraged for trade-off. In addition to the algorithmic advantage of NeuroSA mapping on parallel architecture, neuromorphic hardware like SpiNNaker2 also exhibits energy-to-solution and time-to-solution advantage over CPU as described in the SI Section S5.2, where the near-PE local memory and on-chip random number generator facilitates the acceleration of NeuroSA. The key bottleneck in NeuroSA and other neuromorphic architectures executing random-walk type algorithms is the process of generating the i.i.d. random variables within each neuron. It has been reported that⁶⁹, generating high-quality random noise consumes significant energy and many neuromorphic architectures resort to physical noise (noise intrinsic in devices) as an efficient source of randomness. In our previous works^{48,60,70} we have reported a silicon-compatible device that is capable of producing $\mathcal{O}(1/\log)$ decay required by the FN annealer. The device directly implemented the equivalent circuit shown in Fig. 1e using Fowler–Nordheim tunneling barrier where the current J is determined by single electrons tunneling through the barrier. Future work will investigate how to leverage these discrete single-electron events to produce the other random variables \mathcal{N}_n^B and \mathcal{N}_n^E .

Note that the $\mathcal{O}(1/\log)$ dynamics can either be generated using a physical FN-tunneling device or can be emulated by implementing an FN-tunneling dynamical systems model on digital hardware. In this work, we chose the digital emulation for better scalability. However, a physical FN-tunneling device could also be used as an extrinsic or intrinsic device⁶⁰. The sampling and scaling of the FN-decay can be chosen arbitrarily and is only limited by the resolution of the single-electron tunneling process (quantum uncertainty). This can be mitigated by choosing a larger size device or by modulating the voltage of the device. The time scale of the $\mathcal{O}(1/\log)$ decay can therefore be adjusted to fit the scheduling presented in the algorithm. Note that even with the optimal noise schedule, higher precision sampling and annealing (threshold adjustment) would be required. In SI Fig. S4 we show the degradation in the quality of the solution when a lower precision computation is used for NeuroSA computation. Due to the requirement of higher precision sampling of the FN device, we have resorted to a digital emulation of the FN-tunneling device on the SpiNNaker2 platform. Note that our previously reported FN-devices⁶⁰ could have been used for this purpose, but only in conjunction with analog-to-digital converters with more than 16 bit precision. The scope of this work is to present the algorithmic advancement in developing an asynchronous neuromorphic architecture that can utilize the FN-annealing dynamics.

The NeuroSA architecture opens the possibility of using neuromorphic hardware platforms to find novel solutions by sampling previously unexplored regions of the COP landscape. Given the combinatorial nature of the problem, even a minor improvement in the quality of the solution over the SOTA solution signifies discovering a previously unknown configuration. However, our results suggest that finding such a solution requires a significant number of compute cycles or equivalently a significant expenditure of physical energy. This is evident in Fig. 3f, which plots the number of compute cycles for a unit increase in the solution metric. The trend shows an exponential/sub-exponential growth which highlights the challenge in uncovering new solutions. Because the SA algorithm itself can be slow, the benefits of NeuroSA compared to other polynomial-time COP solvers (like the GW algorithm) lie in the regime where minimal gradient information is available to guide the optimization process, and the algorithm has to resort to a brute force random search. In this region, the search becomes sub-exponential-time or exponential-time and finding the next best solution consumes all of the run-time as was shown in Fig. 3f. However, for NeuroSA, the asymptotic improvement in solution quality is guaranteed due to the FN-tunneling dynamics. This feature

makes NeuroSA stand out among other COP solvers like CIM and SBM. We acknowledge that the proposed ideal neuromorphic ISING machine is a superordinate concept that can use mechanisms other than FN tunneling to achieve annealing and the desired asymptotic convergence properties. This is because there are other stochastic global optimization methods^{52,53} than just simulated annealing which can be used for the neuromorphic ISING architecture. However, when using simulated annealing, according to the results by Hajek⁵⁶ and Geman⁵⁵, the annealing schedule should follow a schedule that is slower than $c/\log(t)$ (or FN tunneling) dynamics to inherit the asymptotic global optimization convergence properties. In fact, the scaling parameter c needs to be selected to achieve the best trade-off between faster convergence and the quality of the solution for a given optimization problem. Also, a schedule that follows $c/\log^{1/2}(t)$ or $c/\log\log(t)$ dynamics would also guarantee asymptotic convergence to the ground state, however, at a much slower rate. Furthermore, we would also point out that NeuroSA, like most COP solvers, is one-shot solver unlike AI/ML inference engines, where the overhead of physical instantiation can be amortized over repetitive runs. Since one of the focus of NeuroSA is to explore new solutions as it asymptotically converges to the ground state, it is expected that NeuroSA implementations would require long run-times. As new/better solutions are produced by the solver they can be read out and used in other applications.

Methods

Asynchronous Ising machine model

QUBO and Ising formulations are interchangeable through a variable transform $s \leftrightarrow \frac{1+s}{2}$, and hence, without any loss of generality, we consider the following optimization problem

$$\min_{\mathbf{s} \in \{-1, +1\}^D} H(\mathbf{s}) = \frac{1}{2} \mathbf{s}^T \mathbf{Q} \mathbf{s} \quad (3)$$

where $\mathbf{s} = [s_1, s_2, \dots, s_D]$ denotes a spin vector comprising of binary optimization variables. Because $s_j^2 = 1, \forall j = 1 \dots D$, Eq. (3) is equivalent to

$$\min_{\mathbf{s} \in \{-1, +1\}^D} H(\mathbf{s}) = \frac{1}{2} \mathbf{s}^T \mathbf{Q} \mathbf{s} \quad \text{with } Q_{ii} = 0 \quad (4)$$

Note the matrix \mathbf{Q} can be symmetrized by $\mathbf{Q} \leftarrow \frac{1}{2}(\mathbf{Q} + \mathbf{Q}^T)$ without changing the solution to Eq. (3). Let the vector \mathbf{s} at time instant n be denoted by \mathbf{s}_n and the change in \mathbf{s} be denoted as $\Delta \mathbf{s}_n$, then

$$\Delta H_n = H(\mathbf{s}_{n-1} + 2\Delta \mathbf{s}_n) - H(\mathbf{s}_{n-1}), \quad (5)$$

where $\Delta \mathbf{s}_n = \{-1, 0, +1\}^D$ and $\Delta s_{j,n} s_{j,n-1} = -1, 0, \forall j = 1 \dots D$ ensures that the spin either flips or remains unchanged. Then,

$$\Delta H_n = \Delta \mathbf{s}_n^T \mathbf{Q} (\mathbf{s}_{n-1} + 2\Delta \mathbf{s}_n) + \mathbf{s}_{n-1}^T \mathbf{Q} \Delta \mathbf{s}_n \quad (6)$$

Using $Q_{ij} = Q_{ji}$,

$$\Delta H_n = 2\Delta \mathbf{s}_n^T \mathbf{Q} (\mathbf{s}_{n-1} + \Delta \mathbf{s}_n) \quad (7)$$

and applying $s_{p,n-1} \Delta s_{p,n} = -1, \forall n, p$ leads to

$$\Delta H_n = 2 \sum_{p \in C} \Delta s_{p,n} \left[\sum_{j \notin C} Q_{pj} s_{j,n} \right], \quad (8)$$

where the set $C = \{i : \Delta s_{i,n} \neq 0\}$ denotes the neurons that do not fire at time-instant n . Solving Eq. (3) involves solving the sequentially sub-problem: $\forall n$, find $\Delta s_{p,n} \in \{-1, 0, +1\}^D$ such that $\sum_{p \in C} \Delta s_{p,n} [\sum_{j \notin C}$

$Q_{pj} s_{j,n}] < 0$, which in itself is a combinatorial problem. By adopting asynchronous firing dynamics, the problem of searching for the set of firing neurons can be simplified. For an asynchronous spiking network, only one of the neurons can emit a spike at any time instant n (due to Poisson statistics), which leads to

$$\Delta H_n = 2\Delta s_{p,n} \left[\sum_{j=1}^D Q_{pj} s_{j,n-1} \right] \quad (9)$$

where we have used $Q_{pp} = 0$. Hence, $\Delta H_n < 0$, if and only if

$$\Delta s_{p,n} \left[\sum_{j=1}^D Q_{pj} s_{j,n-1} \right] < 0. \quad (10)$$

Derivation of NeuroSA's neuron model

In its most general form⁵⁴, a simulated annealing algorithm solves Eq. (3) by accepting or rejecting choices of $\Delta s_{p,n}$ according to

$$\text{Accept } \Delta s_{p,n} : \text{ if } B \exp\left(\frac{-\Delta H_n}{T_n}\right) > u_n, \quad (11)$$

where u_n is a uniformly distributed r.v. between $[0, 1]$, and $B > 1$ is a hyper-parameter, $T_n > 0$ denotes the temperature at time-instant n . Eq. (11) is equivalent to

$$\text{Accept } \Delta s_{p,n} : \text{ if } \Delta H_n < -T_n \log\left(\frac{u_n}{B} + \epsilon\right) \quad (12)$$

or

$$\Delta s_{p,n} \left[\sum_{j=1}^D Q_{pj} s_{j,n-1} \right] < -T_n \mathcal{N}_n^E \quad (13)$$

where $\mathcal{N}_n^E = \log\left(\frac{u_n}{B} + \epsilon\right)$ is an exponentially distributed r.v. We have introduced a small additive term $\epsilon > 0$ to ensure numerical stability when drawing samples with values close to zero. In practice, ϵ is determined by the precision of the hardware platform and hence will be considered a hyperparameter for NeuroSA. Eq. (13) can be written case-by-case as

$$\Delta s_{p,n} = \begin{cases} +1 & \text{if } s_{p,n-1} = -1 \text{ and } -\sum_{j=1}^D Q_{pj} s_{j,n-1} > T_n \mathcal{N}_n^E \\ -1 & \text{if } s_{p,n-1} = +1 \text{ and } \sum_{j=1}^D Q_{pj} s_{j,n-1} > T_n \mathcal{N}_n^E \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

Decomposing the variables differentially as $\Delta s_{p,n} = \Delta s_{p,n}^+ - \Delta s_{p,n}^-$, $s_{p,n} = s_{p,n}^+ - s_{p,n}^-$, $\Delta s_{p,n}^+, \Delta s_{p,n}^-, s_{p,n}^+, s_{p,n}^- > 0$ leads to $s_{p,n}^+ = \sum_{k=1}^n [\Delta s_{p,k}^+ - \Delta s_{p,k}^-]$, $s_{p,n}^- = \sum_{k=1}^n [-\Delta s_{p,k}^+ + \Delta s_{p,k}^-]$. Eq. (14) is therefore equivalent to

$$\Delta s_{p,n}^+ = \begin{cases} 1 & \text{if } s_{p,n-1}^+ = 0 \text{ and } \sum_{k=1}^{n-1} \sum_{j=1}^D Q_{pj} (-\Delta s_{j,k}^+ + \Delta s_{j,k}^-) > T_n \mathcal{N}_n^E \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

which corresponds to the spiking criterion for an ON neuron and

$$\Delta s_{p,n}^- = \begin{cases} 1 & \text{if } s_{p,n-1}^- = 0 \text{ and } \sum_{k=1}^{n-1} \sum_{j=1}^D Q_{pj} (\Delta s_{j,k}^+ - \Delta s_{j,k}^-) > T_n \mathcal{N}_n^E \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

which corresponds to the spiking criterion for an OFF neuron. Introducing a RESET parameter $A \gg |T_n \log(\frac{\mu_n}{B} + \epsilon)|$, Eqs. (15) and (16) are equivalent to the ON neuron model

$$\Delta S_{p,n}^+ = \begin{cases} 1 & \text{if } v_{p,n}^+ > T_n \mathcal{N}_n^E, \\ \text{where } v_{p,n}^+ = v_{p,n-1}^+ + \sum_{j=1}^D Q_{pj} (\Delta S_{j,n-1}^- - \Delta S_{j,n-1}^+) \\ \quad + A \Delta S_{p,n-1}^- - A \Delta S_{p,n-1}^+ & \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

and the OFF neuron model

$$\Delta S_{p,n}^- = \begin{cases} 1 & \text{if } v_{p,n}^- > T_n \mathcal{N}_n^E, \\ \text{where } v_{p,n}^- = v_{p,n-1}^- + \sum_{j=1}^D Q_{pj} (\Delta S_{j,n-1}^+ - \Delta S_{j,n-1}^-) \\ \quad + A \Delta S_{p,n-1}^+ - A \Delta S_{p,n-1}^- & \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

The variables $v_{p,n}^+, v_{p,n}^-$ represent the membrane potentials of the ON–OFF integrate-and-fire neurons at the time instant n . To ensure that all neurons are equally likely to be selected (to satisfy the ergodicity property of SA), we introduce a Bernoulli r.v. for every neuron p as

$$\mathcal{N}_p^B = \begin{cases} 1 & \text{with probability } 1 - \eta \\ 0 & \text{with probability } \eta \end{cases} \quad (19)$$

which leads to the ON neuron model

$$\Delta S_{p,n}^+ = \begin{cases} 1 & \text{if } v_{p,n}^+ > \mu_{p,n}, \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

and the OFF neuron model

$$\Delta S_{p,n}^- = \begin{cases} 1 & \text{if } v_{p,n}^- > \text{manualbreakhskipp} = 12 > \mu_{p,n}, \\ 0 & \text{otherwise,} \end{cases} \quad (21)$$

where $\mu_{n,p} = T_n \mathcal{N}_n^E + A \mathcal{N}_{p,n}^B$ denotes the shared noisy threshold between the p th pair of ON–OFF neurons at time instance n . The ON–OFF construction ensures that $\Delta S_{p,n}^+ \Delta S_{p,n}^- = 0, \forall p, n$, which leads to the following fundamental ON–OFF integrate-and-fire neuron model of NeuroSA which is summarized as: The ON–OFF neuron’s membrane potentials $v_{p,n}^+, v_{p,n}^- \in \mathbb{R}$ evolve as

$$v_{p,n}^+ \leftarrow v_{p,n-1}^+ + \sum_{j=1}^D Q_{pj} (\Delta S_{j,n-1}^- - \Delta S_{j,n-1}^+) + A \Delta S_{p,n-1}^- \quad (22)$$

$$v_{p,n}^- \leftarrow v_{p,n-1}^- + \sum_{j=1}^D Q_{pj} (\Delta S_{j,n-1}^+ - \Delta S_{j,n-1}^-) + A \Delta S_{p,n-1}^+ \quad (23)$$

where $A > 0$ is a constant that represents an excitatory synaptic coupling between the ON and the OFF neurons, as shown in Fig. 1d. The ON and OFF neurons generate a spike when their respective membrane potential exceeds a time-varying noisy threshold $\mu_{p,n}$ according to

$$\Delta S_{p,n}^+ = \begin{cases} 1 & \text{if } v_{p,n}^+ > T_n \mathcal{N}_{p,n}^E + A \mathcal{N}_{p,n}^B, \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

and

$$\Delta S_{p,n}^- = \begin{cases} 1 & \text{if } v_{p,n}^- > T_n \mathcal{N}_{p,n}^E + A \mathcal{N}_{p,n}^B, \\ 0 & \text{otherwise} \end{cases} \quad (25)$$

after which the membrane potentials are RESET by subtraction according to

$$v_{p,n}^+ \leftarrow v_{p,n}^+ - A \Delta S_{p,n}^+ \quad (26)$$

$$v_{p,n}^- \leftarrow v_{p,n}^- - A \Delta S_{p,n}^- \quad (27)$$

Note that the RESET by subtraction is a commonly used mechanism in spiking neural networks and neuromorphic hardware⁷¹. Also, note that the asynchronous RESET of the membrane potential is instantaneous and the spike is represented by a (0,1) binary event.

Dynamical systems model implementing the FN annealer

In ref. 56 it was shown that a temperature annealing schedule of the form

$$T_n \geq \frac{c}{\log(1+n)}. \quad (28)$$

can ensure that the simulated annealing will asymptotically converge to the ground state of the underlying COP. The parameter c in Eq. (28) is chosen to be larger than the depths of all the local minima of COP. The equivalent continuous-time model for the lower-bound in Eq. (28) that produces $T(t)$ is given by

$$T(t) = \frac{T_0}{\log(1 + \frac{t}{C})} \quad (29)$$

where C is a normalizing constant. Differentiating Eq. (29) one obtains the dynamical systems model

$$C \frac{dT}{dt} = -\frac{T^2}{T_0} \exp\left(-\frac{T}{T_0}\right) \quad (30)$$

that generates $T(t)$.

The R.H.S. of Eq. (30) has the form of the current flowing across a Fowler–Nordheim quantum-mechanical tunneling junction and Eq. (30) describes a FN integrator⁶⁰ with a capacitance C . Combining with the expressions of the dynamical threshold in Eqs. (20) and (21), which includes the exponentially distributed and Bernoulli distributed random variables \mathcal{N}_n^E and \mathcal{N}_n^B , leads to the equivalent circuit model of the FN annealer shown in Fig. 1d, e. A discrete-time equation approximating the continuous-time dynamical systems model in Eq. (30) was implemented on the CPU or the SpiNNaker2 platform. The sampling period in the discrete-time model was chosen by adjusting C and to fit the optimal scheduling requirements of the NeuroSA algorithm. For all experiments, $C = 8 \times 10^4$, and $T_0 = 0.3125$. The mean of the random variable $\mathcal{N}_{p,n}^E$ was chosen to be -0.916 .

Acceleration of NeuroSA on synchronous and clocked systems

While the ideal implementation of NeuroSA architecture requires a fully asynchronous architecture, most neuromorphic accelerators are either fully clocked and use address-event-routing-based packet switching, such as HiAER-spike, or employ globally asynchronous interrupt-driven units that are locally clocked (synchronous), such as SpiNNaker2. For these clocked systems, NeuroSA can be efficiently implemented by exploiting the mutual independence and i.i.d. properties of the r.v.s. \mathcal{N}_n^E and \mathcal{N}_n^B . The SI Section S4 describes the pseudocode for CPU and SpiNNaker2 implementations. In the NeuroSA architecture, each neuron determines its spiking behavior solely from its internal parameters, i.e. the membrane potential, neuron state, etc. Therefore, $\mathcal{N}_{p,n}^E$ needs to be distinct and local to each neuron in the system. On the other hand, the ergodicity of the optimization process can be enforced through a global arbiter. We decouple the Bernoulli

noise from the noisy threshold such that only the decision threshold $\mu_{p,n}^* = T_n \wedge_{p,n}^E$ is applied to each neuron. In this case, multiple spikes may occur at each simulation step. All the neurons that emit spikes synchronously are referred to as active neurons. Out of all active neurons, only one gets selected by the global arbiter and propagated to other neurons, while the remaining spikes are discarded. This inhibitive firing dynamics ensures that at most one spike is transmitted and processed, which satisfies the asynchronous firing requirement as shown in Eq. (9). The global arbiter is implemented differently across the neuromorphic hardware that we have tested on, with the detailed implementation documented in SI Section S4 for CPU (software) implementation, and Section S5 for SpiNNaker2 implementation.

Generation of network PCA trajectories

To demonstrate and visualize the evolution of the network dynamics for a large problem, we used Principal component analysis (PCA) to perform dimensionality reduction on the population dynamics similar to a procedure reported in literature^{63,72,73}. In NeuroSA, the population spiking activity indicates changes in the neuronal states, the attractor dynamics in proximity to a local/global minimum, and the escape mechanisms for exploring the state space. As shown in Fig. 1h, the spikes across the neuronal ensembles are binned within a predefined time window to produce a real-valued vector. The time window is then shifted with some pre-defined overlap to produce a sequence of real-valued vectors. PCA is then performed over all the real-valued vectors and only the principal vectors with largest eigenvalues are chosen. The real-valued vector sequence is then projected onto the three principal components resulting in 3D trajectories shown in Figs. 1i and 2d.

Estimation of empirical probability density function

For the MAX-CUT and MIS experiments, NeuroSA was run on each graph for 5 times. The results in each run were normalized with respect to the SOTA and then grouped as a histogram. The probability density function (pdf) was generated using `ksdensity`, the built-in MATLAB function.

Data availability

The COP solutions, the algorithm execution time, the power and the energy data for the CPU and SpiNNaker2 implementation of NeuroSA generated in this study are available at <https://github.com/aimlab-wustl/neuroSA>.

Code availability

The specific MATLAB and Python codes used in simulation/emulation studies on the CPU platform are available at <https://github.com/aimlab-wustl/neuroSA>. Since the SpiNNaker2 stack is still under development, and the system is proprietary, the software implementation on SpiNNaker2 will not be made available publicly but will be available upon requests.

References

- Barahona, F. On the computational complexity of ising spin glass models. *J. Phys. A: Math. Gen.* **15**, 3241 (1982).
- Lucas, A. Ising formulations of many np problems. *Front. Phys.* **2**, <https://doi.org/10.3389/fphy.2014.00005> (2014).
- Mohseni, N., McMahon, P. L. & Byrnes, T. Ising machines as hardware solvers of combinatorial optimization problems. *Nat. Rev. Phys.* **4**, 363–379 (2022).
- Hamerly, R. et al. Experimental investigation of performance differences between coherent ising machines and a quantum annealer. *Sci. Adv.* **5**, 0823 (2019).
- Tanahashi, K., Takayanagi, S., Motohashi, T. & Tanaka, S. Application of ising machines and a software development for ising machines. *J. Phys. Soc. Jpn.* **88**, 061010 (2019).
- King, A. D. et al. Coherent quantum annealing in a programmable 2000 qubit ising chain. *Nat. Phys.* **18**, 1324–1328 (2022).
- Mwamsojo, N., Lehmann, F., Merghem, K., Benkelfat, B.-E. & Frignac, Y. Optoelectronic coherent ising machine for combinatorial optimization problems. *Opt. Lett.* **48**, 2150–2153 (2023).
- Honjo, T. et al. 100,000-spin coherent ising machine. *Sci. Adv.* **7**, 0952 (2021).
- Graber, M. & Hofmann, K. An enhanced 1440 coupled CMOS oscillator network to solve combinatorial optimization problems. In *2023 IEEE 36th International System-on-Chip Conference (SOCC)* (ed. Becker, J.) 1–6 (IEEE, 2023).
- Maher, O. et al. A CMOS-compatible oscillation-based VO₂ Ising machine solver. *Nat. Commun.* **15**, 3334 (2024).
- Cai, F. et al. Power-efficient combinatorial optimization using intrinsic noise in memristor hopfield neural networks. *Nat. Electron.* **3**, 409–418 (2020).
- Jiang, M., Shan, K., He, C. & Li, C. Efficient combinatorial optimization by quantum-inspired parallel annealing in analogue memristor crossbar. *Nat. Commun.* **14**, <https://doi.org/10.1038/s41467-023-41647-2> (2023).
- Fahimi, Z., Mahmoodi, M., Nili, H., Polishchuk, V. & Strukov, D. Combinatorial optimization by weight annealing in memristive Hopfield networks. *Sci. Rep.* **11**, <https://doi.org/10.1038/s41598-020-78944-5> (2021).
- Isakov, S. V., Zintchenko, I. N., Rønnow, T. F. & Troyer, M. Optimised simulated annealing for ising spin glasses. *Comput. Phys. Commun.* **192**, 265–271 (2015).
- Kihara, Y. et al. A new computing architecture using ising spin model implemented on fpga for solving combinatorial optimization problems. In *2017 IEEE 17th International Conference on Nanotechnology (IEEE-NANO)* 256–258 (IEEE, 2017).
- Yamaoka, M. et al. A 20k-spin ising chip to solve combinatorial optimization problems with cmos annealing. *IEEE J. Solid-State Circuits* **51**, 303–309 (2015).
- Okuyama, T., Yoshimura, C., Hayashi, M. & Yamaoka, M. Computing architecture to perform approximated simulated annealing for ising models. In *2016 IEEE International Conference on Rebooting Computing (ICRC)* 1–8 (IEEE, 2016).
- Kadowaki, T. & Nishimori, H. Quantum annealing in the transverse ising model. *Phys. Rev. E* **58**, 5355 (1998).
- Das, A. & Chakrabarti, B. K. Colloquium: quantum annealing and analog quantum computation. *Rev. Mod. Phys.* **80**, 1061 (2008).
- Farhi, E. et al. A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. *Science* **292**, 472–475 (2001).
- Albash, T. & Lidar, D. A. Adiabatic quantum computation. *Rev. Mod. Phys.* **90**, 015002 (2018).
- Goto, H., Tatsumura, K. & Dixon, A. R. Combinatorial optimization by simulating adiabatic bifurcations in nonlinear hamiltonian systems. *Sci. Adv.* **5**, 2372 (2019).
- Grimaldi, A. et al. Evaluating spintronics-compatible implementations of ising machines. *Phys. Rev. Appl.* **20**, 024005 (2023).
- Singh, N.S. et al. Cmos plus stochastic nanomagnets enabling heterogeneous computers for probabilistic inference and learning. *Nat. Commun.* **15**, <https://doi.org/10.1038/s41467-024-46645-6> (2024).
- Kang, M., Lee, Y. & Park, M. Energy efficiency of machine learning in embedded systems using neuromorphic hardware. *Electronics* **9**, 1069 (2020).
- Ivanov, D., Chezhegov, A. & Larionov, D. Neuromorphic artificial intelligence systems. *Front. Neurosci.* **16**, 959626 (2022).
- Furber, S. B., Galluppi, F., Temple, S. & Plana, L. A. The spinnaker project. *Proc. IEEE* **102**, 652–665 (2014).
- Park, J., Yu, T., Joshi, S., Maier, C. & Cauwenberghs, G. Hierarchical address event routing for reconfigurable large-scale neuromorphic

- systems. *IEEE Trans. Neural Netw. Learn. Syst.* **28**, 2408–2422 (2017).
29. Davies, M. et al. Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* **38**, 82–99 (2018).
30. Mayr, C., Hoepfner, S. & Furber, S. Spinnaker 2: a 10 million core processor system for brain simulation and machine learning. arXiv preprint arXiv:1911.02385 (2019).
31. Van Albada, S. J. et al. Performance comparison of the digital neuromorphic hardware spinnaker and the neural network simulation software nest for a full-scale cortical microcircuit model. *Front. Neurosci.* **12**, 309524 (2018).
32. Rhodes, O. et al. Real-time cortical simulation on neuromorphic hardware. *Philos. Trans. R. Soc. A* **378**, 20190160 (2020).
33. Liu, C. et al. Memory-efficient deep learning on a spinnaker 2 prototype. *Front. Neurosci.* **12**, 416510 (2018).
34. Lin, C.-K. et al. Programming spiking neural networks on intel's loihi. *Computer* **51**, 52–61 (2018).
35. Gonzalez, H. A. et al. SpiNNaker2: a large-scale neuromorphic system for event-based and asynchronous machine learning. Machine Learning with New Compute Paradigms Workshop at NeurIPS (MLNPCP) (2023).
36. Shankar, S. Energy estimates across layers of computing: from devices to large-scale applications in machine learning for natural language processing, scientific computing, and cryptocurrency mining. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)* 1–6 (IEEE, 2023).
37. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl Acad. Sci. USA* **79**, 2554–2558 (1982).
38. Friston, K. The free-energy principle: a unified brain theory? *Nat. Rev. Neurosci.* **11**, 127–138 (2010).
39. Chialvo, D. R. Emergent complex neural dynamics. *Nat. Phys.* **6**, 744–750 (2010).
40. Sompolinsky, H., Crisanti, A. & Sommers, H.-J. Chaos in random neural networks. *Phys. Rev. Lett.* **61**, 259 (1988).
41. McDonnell, M. D. & Ward, L. M. The benefits of noise in neural systems: bridging theory and experiment. *Nat. Rev. Neurosci.* **12**, 415–425 (2011).
42. Schneidman, E., Freedman, B. & Segev, I. Ion channel stochasticity may be critical in determining the reliability and precision of spike timing. *Neural Comput.* **10**, 1679–1703 (1998).
43. Branco, T., Staras, K., Darcy, K. J. & Goda, Y. Local dendritic activity sets release probability at hippocampal synapses. *Neuron* **59**, 475–485 (2008).
44. Jaeger, H. & Haas, H. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *science* **304**, 78–80 (2004).
45. Hoppensteadt, F. C. & Izhikevich, E. M. Oscillatory neurocomputers with dynamic connectivity. *Phys. Rev. Lett.* **82**, 2983 (1999).
46. Fischer, A. & Igel, C. An introduction to restricted Boltzmann machines. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: Proc. of the 17th IberoAmerican Congress, CIARP 2012* (eds Alvarez, L., Mejail, M., Gomez, L. & Jacobo, J.) 14–36 (Springer, 2012).
47. Amin, M. H., Andriyash, E., Rolfe, J., Kulchitsky, B. & Melko, R. Quantum Boltzmann machine. *Phys. Rev. X* **8**, 021050 (2018).
48. Mehta, D., Rahman, M., Aono, K. & Chakrabarty, S. An adaptive synaptic array using fowler-nordheim dynamic analog memory. *Nat. Commun.* **13**, <https://doi.org/10.1038/s41467-022-29320-6> (2022).
49. Smith, J. D. et al. Neuromorphic scaling advantages for energy-efficient random walk computations. *Nat. Electron.* **5**, 102–112 (2022).
50. Raghavendra, P. Optimal algorithms and inapproximability results for every CSP? In *Proc. 40th Annual ACM Symposium on Theory of Computing, STOC'08* 245–254 (Association for Computing Machinery, 2008).
51. Khot, S. On the unique games conjecture (invited survey). In *2010 IEEE 25th Annual Conference on Computational Complexity* 99–121 (IEEE, 2010).
52. Theilman, B. H. & Aimone, J. B. Goemans–Williamson maxcut approximation algorithm on loihi. In *NICE '23*, (eds Kudithipudi, D., Frenkel, C., Cardwell, S. & Aimone, J. B.) 1–5 (Association for Computing Machinery, 2023).
53. Theilman, B.H. et al. Stochastic neuromorphic circuits for solving MAXCUT. In *2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS)* (ed. O'Connor, L.) 779–787 (IEEE, 2023).
54. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science* **220**, 671–680 (1983).
55. Geman, S. & Geman, D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-6**, 721–741 (1984).
56. Hajek, B. Cooling schedules for optimal annealing. *Math. Oper. Res.* **13**, 311–329 (1988).
57. Lucas, A. Ising formulations of many np problems. *Front. Phys.* **2**, 74887 (2014).
58. Goemans, M. X. & Williamson, D. P. Improved approximation algorithms for maximum cut and satisfiability problems using semi-definite programming. *J. ACM* **42**, 1115–1145 (1995).
59. Fowler, R. H. & Nordheim, L. Electron emission in intense electric fields. *Proc. R. Soc. Lond. Ser. A* **119**, 173–181 (1928).
60. Zhou, L. & Chakrabarty, S. Self-powered timekeeping and synchronization using fowler-nordheim tunneling-based floating-gate integrators. *IEEE Trans. Electron Devices* **64**, 1254–1260 (2017).
61. Matsuda, Y. *Benchmarking the MAX-CUT problem on the Simulated Bifurcation Machine* —medium.com <https://medium.com/toshiba-sbm/benchmarking-the-max-cut-problem-on-the-simulated-bifurcation-machine-e26e1127c0b0>.
62. Yik, J. et al. The neurobench framework for benchmarking neuromorphic computing algorithms and systems. *Nat. Commun.* **16**, 1545 (2025).
63. Stopfer, M., Jayaraman, V. & Laurent, G. Intensity versus identity coding in an olfactory system. *Neuron* **39**, 991–1004 (2003).
64. Layden, D. et al. Quantum-enhanced markov chain monte carlo. *Nature* **619**, 282–287 (2023).
65. Ye, Y. *The Gset Dataset* <https://web.stanford.edu/yyyy/yyyy/Gset/> (2003).
66. Dehmer, M. & Mowshowitz, A. A history of graph entropy measures. *Inf. Sci.* **181**, 57–78 (2011).
67. Holland, P. W. & Leinhardt, S. Transitivity in structural models of small groups. *Comput. Group Stud.* **2**, 107–124 (1971).
68. Ozaeta, A., Dam, W. & McMahon, P. L. Expectation values from the single-layer quantum approximate optimization algorithm on ising problems. *Quantum Sci. Technol.* **7**, 045036 (2022).
69. Pierog, T., Karpenko, I., Katzy, J. M., Yatsenko, E. & Werner, K. Epos lhc: test of collective hadronization with data measured at the cern large hadron collider. *Phys. Rev. C* **92**, 034906 (2015).
70. Mehta, D., C.S., Aono, A. A self-powered analog sensor-data-logging device based on Fowler–Nordheim dynamical systems. *Nat. Commun.* **11**, <https://doi.org/10.1038/s41467-020-19292-w> (2020).
71. Rueckauer, B., Lungu, I.-A., Hu, Y., Pfeiffer, M. & Liu, S.-C. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Front. Neurosci.* **11**, <https://doi.org/10.3389/fnins.2017.00682> (2017).
72. Friedrich, R. W. & Laurent, G. Dynamic optimization of odor representations by slow temporal patterning of mitral cell activity. *Science* **291**, 889–894 (2001).

73. Gangopadhyay, A. & Chakrabarty, S. A sparsity-driven back-propagation-less learning framework using populations of spiking growth transform neurons. *Front. Neurosci.* **15**, 715451 (2021).

Acknowledgements

This work is supported in part by research grants from the US National Science Foundation: ECCS:2332166 and FET:2208770. M.A., H.G., G.L., M.S., and C.M. would like to acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the program of “Souverän. Digital. Vernetzt.”. Joint project 6G-life, project identification number: 16KISK001K. M.A., H.G., G.L., M.S., and C.M. acknowledge the EIC Transition under the “SpiNNaker” project (grant number 101112987). M.A., H.G., G.L., M.S., and C.M. also acknowledge the Horizon Europe project “PRIMI” (Grant number 101120727). G.U. acknowledges a contribution from the Italian National Recovery and Resilience Plan (NRRP), M4C2, funded by the European Union-NextGenerationEU (Project IR0000011, CUP B51E22000150006, “EBRAINS-Italy”). S.S. acknowledges Department of Energy’s Office of Science contract DE-AC02-76SF00515 with SLAC through an Annual Operating Plan agreement WBS 2.1.0.86 from the Office of Energy Efficiency and Renewable Energy’s Advanced Manufacturing and Materials Technology Office and the institutional support from SLAC National Laboratory.

Author contributions

Z.C., Z.X., M.A., J.L., O.O., A.M., N.D., C.H., S.B., J.E., R.P., G.U., A.G.A., S.S., C.M., G.C., and S.C. participated in a workgroup titled Quantum-inspired Neuromorphic Systems at the Telluride Neuromorphic and Cognitive Engineering (TNCE) workshop in 2023, and the outcomes from the workgroup have served as the motivation for this work. S.C. formulated the asynchronous ON-OFF neuron model with the FN-annealer; Z.C. and S.C. designed the NeuroSA experiments; Z.C. benchmarked NeuroSA on different MAX-CUT graphs; Z.X. implemented the first version of SA algorithm; S.C., Z.C., J.L. and G.C. proposed the use of spike events to reduce communication bottleneck in NeuroSA; M.A., H.G., C.M., M.S., G.L., and Z.C. implemented NeuroSA on SpiNNaker2; M.A., M.S., and G.L. optimized SpiNNaker2 for MAX-CUT benchmarks; All authors/co-authors contributed to proof-reading and writing of the manuscript.

Competing interests

SpiNNaker2 is a neuromorphic hardware accelerator platform by SpiNNcloud Systems, a commercial entity with whom M.A., H.G., M.S., G.L., and C.M. have affiliations and financial interests. S.C. is named as

an inventor on U.S. and international patents associated with FN-based dynamical systems, and the rights to the intellectual property are managed by Washington University in St. Louis. The remaining authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41467-025-58231-5>.

Correspondence and requests for materials should be addressed to Shantanu Chakrabarty.

Peer review information *Nature Communications* thanks James Aimone, Tetsufumi Tanamoto and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at <http://www.nature.com/reprints>

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

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025

¹Department of Electrical and Systems Engineering, Washington University in St. Louis, One Brookings Drive, St. Louis, MO 63130, USA. ²SpiNNcloud Systems GmbH, Freiburgerstr. 37, Dresden 01067, Germany. ³Department of Bioengineering, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093, USA. ⁴Department of Electrical and Electronic Engineering, Imperial College London, Exhibition Rd, London SW7 2AZ, UK. ⁵International Centre for Neuromorphic Engineering, Western Sydney University, Penrith, Second Ave, Kingswood 2747 NSW, Australia. ⁶Biocomputation Group, University of Hertfordshire, Exhibition Rd, London SW7 2AZ, UK. ⁷Department of Physics, University of California, Berkeley, University Avenue and Oxford St, Berkeley, CA 94720, USA. ⁸Redwood Center for Theoretical Neuroscience and Helen Wills Neuroscience Institute, University of California, Berkeley, University Avenue and Oxford St, Berkeley, CA 94720, USA. ⁹Chair of Highly-Parallel VLSI-Systems and Neuro-Microelectronics, Technische Universität Dresden, Mommsenstraße 12, Dresden 01069, Germany. ¹⁰Department of Electrical and Computer Engineering, University of California, Santa Cruz, 1156 High Street, Santa Cruz, CA 95064, USA. ¹¹Politecnico di Torino, Corso Duca degli Abruzzi, 24, Torino 10129, Italy. ¹²Department of Electrical and computer engineering, Johns Hopkins University, 3400 N. Charles Street, Baltimore, MD 21218, USA. ¹³SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025, USA. ¹⁴Materials Science and Engineering, Stanford University, 450 Jane Stanford Way, Stanford, CA 94305, USA. ¹⁵Scads.AI: Center for Scalable Data Analytics and Artificial Intelligence, Strehlener Street 12, 14, Dresden 01069, Germany. ✉ e-mail: shantanu@wustl.edu