A Spiking Neuron and Population Model based on the Growth Transform Dynamical System

Ahana Gangopadhyay[1], Darshit Mehta[2], and Shantanu Chakrabartty[1]

[1]Department of Electrical and Systems Engineering, Washington University in St. Louis
[2]Department of Biomedical Engineering, Washington University in St. Louis

Abstract

This paper introduces a new spiking neuron and population model that allows for independent control over three neuro-dynamical properties: (a) control over the steady-state population dynamics that encodes the minimum of an exact network energy functional; (b) control over the shape of the action potentials generated by individual neurons in the network without affecting the network minimum; and (c) control over spiking statistics and transient population dynamics without affecting the network minimum or the shape of the action potentials. At the core of the proposed model are different variants of Growth Transform dynamical systems that produce stable and interpretable population dynamics, irrespective of the network size and the type of neuronal connectivity (inhibitory or excitatory). In this paper, we present several examples where the proposed model has been configured to produce different types of single-neuron dynamics as well as population dynamics. In one such example, the network is shown to adapt such that it encodes the steady-state solution using a reduced number of spikes. In this paper we use this network to construct a spiking associative memory that uses fewer spikes compared to conventional architectures, while maintaining high recall accuracy at high memory loads.

Keywords— Spiking neuron model, Growth Transforms, energy-minimization, dynamical system, network model, neural dynamics, associative memory, adaptation.

1 Introduction

The nervous system - from the cellular level to neural ensembles - has been studied extensively within the context of dynamical systems [1], and modeled as a set of differential equations that govern the temporal evolution of its state variables. For a single neuron, the state variables are usually its membrane potential and the conductances of ion channels that mediate changes in the membrane potential via flux of ions across the cell membrane. A vast body of literature, ranging from the classical Hodgkin-Huxley model [2], FitzHugh-Nagumo model [3], Izhikevich model [4] to simpler integrate-and-fire models [5], treats the problem of single-cell excitability at various levels of detail and biophysical plausibility. Individual neuron models are then connected through synapses, bottom-up, to form large-scale spiking neural networks.

An alternative to this bottom-up approach is a top-down approach that treats the process of spike generation and neural representation in the context of minimizing some measure of network energy. The rationale for this approach is that physical processes occurring in nature have a tendency to self-optimize towards a minimum-energy state. This principle has been used to design neuromorphic systems where the state of a neuron in the network is assumed to be either binary in nature (spiking or not spiking) [6], or replaced by its average firing rate [7]. Recently, energy-based attractor networks have been used to design spiking neural networks

through phase-to-timing mappings in complex state space [8]. Energy has also been modeled as an external resource to constrain and control neural activity [9]. However, in all of these approaches, the energy functionals have been defined with respect to some statistical measure of neural activity, for example spike rates, instead of continuous-valued neuronal variables like the membrane potential. As a result in these models, it is difficult to independently control different neuro-dynamical parameters, for example the shape of the action-potential, bursting activity or adaptation in neural activity, without affecting the network solution.

In [10], we proposed a model of a Growth Transform (GT) neuron which reconciled the bottom-up and top-down approaches such that the dynamical and spiking responses were derived directly from a network objective or an energy functional. It was shown in [10] that a network of GT neurons can solve binary classification tasks while producing stable and unique neural dynamics (for example, noise-shaping, spiking and bursting) that could be interpreted using a classification margin. However, in the previous formulation, all of these neuro-dynamical properties were directly encoded into the network energy function. As a result, the formulation did not allow independent control and optimization of different neuro-dynamics. In this paper, we address these limitations by proposing a novel GT spiking neuron and population model, along with a neuromorphic framework, according to the following steps:

- We first remap the synaptic interactions in a standard spiking neural network in a manner that the solution (steady-state attractor) could be encoded as a first-order condition of a convex optimization problem. We show that this convex network objective function or network energy functional can be interpreted as the total extrinsic power required by the network to operate, and hence a metric to be minimized based on evolutionary principles.

- We then introduce a dynamical system model based on Growth Transforms that evolves the network towards this steady-state attractor. The use of Growth Transforms ensures that the neuronal states (membrane potentials) involved in the optimization are always bounded and that each step in the evolution is guaranteed to reduce the network energy.

- We then introduce the ability to control the shape of the action potential by incorporating a gradient discontinuity in the regularization function of the network energy functional. The gradient discontinuity will be chosen such that the local convexity of the energy functional and hence the location of the steady-state attractor is preserved.

- Finally, we use the properties of Growth Transforms to generalize the model to a continuous-time dynamical system. The formulation will then allow for modulating the spiking and the population dynamics across the network without affecting network convergence to the steady-state attractor.

We show that the proposed framework can be used to implement a network of coupled neurons that can exhibit memory, global adaptation, and other interesting population dynamics under different initial conditions and based on different network states. As a representative example, we use the model to design a spiking associative memory network and show how the network can recall a large number of patterns with high accuracy while using fewer spikes than traditional associative memory networks. This paper is also accompanied by a publicly available software implementing the proposed model [11] using MATLAB©. Users can experiment with different inputs and network parameters to explore and create other unique dynamics than what has been reported in this paper. In the future, we envision that the model could be extended to incorporate spike-based learning within an energy-minimization framework similar to the framework used in traditional machine learning models [12]. This could be instrumental in bridging the gap between neuromorphic algorithms and traditional energy-based machine learning models.

2 Methods

For the rest of the paper, we will follow the mathematical notations as summarized below.

Notations

x	Real scalar variable
$\boldsymbol{x}$	Real-valued vector with x_i as its i-th element
$\boldsymbol{X}$	Real-valued matrix with X_{ij} as the element at the i-th row and the j-th column
$x_{i,n}$	i-th element of real-valued vector $\boldsymbol{x}$ at the n-th time step
$x_i(t)$	i-th element of real-valued vector $\boldsymbol{x}$ at time t
$\mathcal{E}_n[x_{i,n}]$ or $\bar{x}_i$	Expectation of the i-th element of real-valued vector $\boldsymbol{x}$, i.e., $\lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} x_{i,n}$
$\mathcal{E}_t[x_i(t)]$	Expectation of the i-th element of real-valued vector $\boldsymbol{x}$, i.e., $\lim_{T\to\infty} \frac{1}{T}\int_{t=0}^{T} x_i(t)dt$
$\mathcal{R}^M$	Vector space spanned by M-dimensional real vectors
$\|\|.\|\|_p$	p-th norm of a vector, defined as $(\sum_i \|x_i\|^p)^{1/p}$
$\|.\|$	Absolute value of a scalar
$\mathcal{H}(.)$	Scalar-valued function of a vector
$\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}}$	Gradient vector $[\frac{\partial \mathcal{H}}{\partial x_1}, \frac{\partial \mathcal{H}}{\partial x_2}, ..., \frac{\partial \mathcal{H}}{\partial x_M}]^T$

2.1 Remapping synaptic interactions in a standard spiking network

In this section, we derive the network energy functional by remapping the synaptic interactions of a standard spiking neural network and then propose a Growth Transform based dynamical system for minimizing this objective. In generalized threshold models like the Spike Response Model [13], the membrane voltage is given using response kernels that accurately model the post-synaptic responses due to pre-synaptic input spikes, external driving currents and the shape of the spike - the latter term being also used to model refractoriness. However, in simpler adaptations of spiking neuron models, the spike shape is often disregarded, and the membrane potentials are written as simple linear post-synaptic integrations of input spikes and external currents [14, 15]. Such a spiking neural network model is shown in Fig. 1(a), where v_i represents an inter-cellular membrane potential corresponding to the neuron i in a network of M neurons. The i-th neuron receives a spike $\theta(v_j)$ (which may be current waveforms like exponential decays or α-functions [16, 17] or simply a digital bit [15]) from the j-th neuron and is modulated by a synapse whose strength or weight is denoted by W_{ij}. In this model, y_i represents the depolarization due to any external current stimulus. In this case, and under standard assumptions, the activity of the i-th neuron can be expressed by the following equation (detailed justification provided in Appendix A)

$$r_i = \Phi(\sum_{j=1}^{M} W_{ij} s_j + \bar{y}_i). \tag{1}$$

Here, $r_i = \mathcal{E}_n[\theta(v_{i,n})]$ denotes the average spiking rate of the i-th neuron over a sufficiently large window of time, s_j is the average or filtered spike-train of the j-th neuron and $\bar{y}_i$ denotes the average input stimulus received by the i-th neuron. Note that r_i, s_i and $\bar{y}_i$ are all functions of time. The non-linear function $\Phi(.)$ models the rate-saturation and the effect of leaky integration (see Appendix A for more details). To map this into a network objective, one needs to find an energy functional for which (1) is the first-order condition. Unfortunately, (1) is a function of the activity variables r_i, estimating which requires windowing over large periods of time, and also eradicates the possibility of utilizing information in spike-times or instantaneous changes

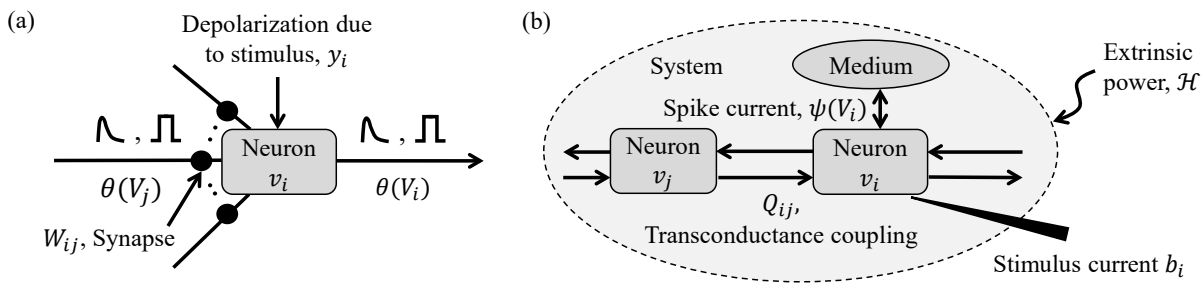

Figure 1: (a) Simple but generalized model of a spiking neural network; (b) Compartmental network model obtained after remapping.

in spike dynamics. Moreover, the non-linearity inherently present in s_j makes it difficult to derive an exact form of the network energy functional. Instead, an ideal first-order condition would be a form that can be written directly in terms of the membrane potentials v_i and the instantaneous depolarization y_i due to external current as

$$v_i = \sum_{j=1}^{M} W_{ij}\Psi(v_j) + y_i, \tag{2}$$

where $\Psi(.)$ denotes the filtered spike current and is a general non-linear function of the membrane potential. For instance, $\Psi(v_j)$ could be a simple spiking function that is positive only when the voltage v_j exceeds a threshold. Note that in (2), the filter $\Psi(.)$ implicitly depends on the pre-synaptic spike-times through the pre-synaptic membrane voltage v_j. Also, to ensure that the re-mapping procedure is stable, we will enforce the following constraints on the membrane potentials

$$|v_i| \leq 1, \;\; \forall i = 1, ..., M. \tag{3}$$

Note that in biological neural networks, the membrane potentials are also bounded [18] and the unity constraint in 3 has an effect similar to the saturation non-linearity $\Phi(.)$ in 1.

To reduce notational clutter, we will re-write (2) in a matrix form as

$$\boldsymbol{v} = \boldsymbol{W}\Psi(\boldsymbol{v}) + \boldsymbol{y}. \tag{4}$$

Again, due to the non-linear function $\Psi(.)$ in (4), it is difficult to derive an exact network energy functional corresponding to (4). However, if we assume that the synaptic weight matrix $\boldsymbol{W}$ is invertible, we can re-write Eq. (4) as

$$\Psi(\boldsymbol{v}) = \boldsymbol{W}^{-1}\boldsymbol{v} - \boldsymbol{W}^{-1}\boldsymbol{y}, \tag{5}$$

$$\Psi(\boldsymbol{v}) = -\boldsymbol{Q}\boldsymbol{v} + \boldsymbol{b}, \tag{6}$$

where $\boldsymbol{Q} = -\boldsymbol{W}^{-1}$, and $\boldsymbol{b} = -\boldsymbol{W}^{-1}\boldsymbol{y}$ is the effective external current stimulus. Note that in case $\boldsymbol{W}$ is not invertible, $\boldsymbol{W}^{-1}$ could represent a pseudo-inverse. For the i-th neuron, (6) is equivalent to

$$\Psi(v_i) = -\sum_{j=1}^{M} Q_{ij}v_j + b_i, \tag{7}$$

subject to the bound constraint $|v_i| \leq 1$. This re-mapping allows us to write Eq. (7) as a first-order condition for minimization of an exact network objective function or energy functional

which is given by

$$\min_{|v_i|\leq 1} \mathcal{H}(v_i) = \min_{|v_i|\leq 1} \ \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} Q_{ij}v_jv_i - \sum_{i=1}^{M} b_iv_i + \sum_{i=1}^{M}\int \Psi(v_i)dv_i. \tag{8}$$

The network energy functional $\mathcal{H}(.)$ in (8) also admits a physical interpretation, as shown in Fig. 1(b). Each neuron i receives a voltage input from the neuron j through a synapse that can be modeled by a transconductance Q_{ij} [19]. As shown in Fig. 1(a), the neuron i also receives an electrical current stimuli b_i and exchanges voltage-dependent ionic-current with its medium, denoted by $\Psi(v_i)$. Then, the function $\mathcal{H}(.)$ in Eq. (8) represents the extrinsic (or metabolic) power supplied to the network, comprising the following three components: (a) Power dissipation due to coupling between neurons; (b) Power injected to or extracted from the system as a result of external stimulation; and (c) Power dissipated due to neural responses.

Note that the network objective in (8) bears similarity with the Ising Hamiltonians used in Hopfield networks [20], Boltzmann machines [21] or spin-glass models [22]. However, there are two key differences:

- Our proposed energy functional is a function of continuous-valued, time-varying neural variables (membrane potential), where the synaptic weights are not necessarily symmetric;
- The energy functional consists of an additional term $\int \Psi(.)dv$, which models the effect of the transmembrane current, and acts as a regularizer. This regularization function has been used in our framework to incorporate the spiking dynamics.

2.2 Neuron models using Growth Transform dynamical systems

In order to solve the energy minimization problem given in (8) under the constraints given in (3), we propose a dynamical system based on growth transforms - a fixed-point multiplicative update used for optimizing polynomial functions under linear and/or bound constraints [23, 24]. The dynamical system evolves over time to reach an optimal point of the energy landscape over a constraint manifold. The summary of the proposed model is presented in Table 1 and the detailed derivation is provided in Appendix B.

Table 1: Discrete-time Growth Transform dynamical system (Proof in Appendix B)

Proposition I. Let $\mathcal{H}(\boldsymbol{v}) : \mathcal{R}^M \rightarrow \mathcal{R}$ be a Lipschitz-continuous function of $\boldsymbol{v} = \{v_1, v_2, .., v_M\}$, and let $\lambda > ||\frac{\partial \mathcal{H}}{\partial \boldsymbol{v}}||_0$, be a hyper-parameter. Then for $|v_{i,0}| \leq 1 \ \forall i$, the discrete-time dynamical system

$$v_{i,n+1} \leftarrow \frac{-\frac{\partial \mathcal{H}}{\partial v_{i,n}} + \lambda v_{i,n}}{-v_{i,n}\frac{\partial \mathcal{H}}{\partial v_{i,n}} + \lambda}, \ \ i = 1, ..., M \tag{9}$$

satisfies the following criteria for all time-indices n:

$$(a) \ \ |v_{i,n}| \leq 1 \ \ \forall i; \tag{10}$$

$$(b) \ \ \mathcal{H}(\boldsymbol{v}_{n+1}) \leq \mathcal{H}(\boldsymbol{v}_n); \ \text{ and} \tag{11}$$

$$(c) \ \ \mathcal{E}_n\Big[(1 - v_{i,n}v_{i,n+1})\frac{\partial \mathcal{H}}{\partial v_{i,n}}\Big] \rightarrow 0 \ \ \forall i. \tag{12}$$

Hence for a continuously differentiable function $\mathcal{H}(.)$, the dynamical system evolves over time to reach a local minimum of $\mathcal{H}(.)$ within the domain defined by the bound constraints.

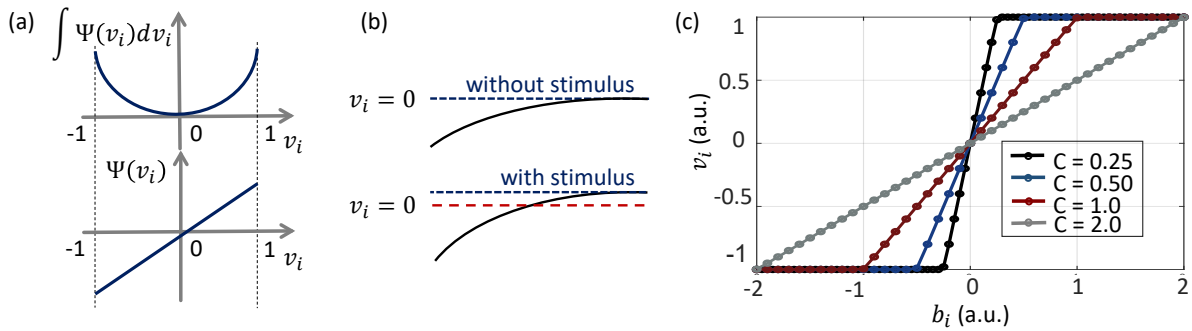

Figure 2: (a) A quadratic form of the function $\Psi(v_i)dv_i$ and its corresponding derivative $\Psi(.)$; (b) Time-evolution of the response v_i of a single neuron in the non-spiking model in the absence and presence of external stimulus; (c) Input-output characteristics of the non-spiking neuron model for different values of C.

2.2.1 Growth Transform non-spiking neuron model

To show the dynamics resulting from substituting (8) in the Growth Transform dynamical system described in (9), we first consider $\Psi(.)$ such that the integral $\int \Psi(v_i)dv_i$ over $[-1, 1]$ is a smooth and convex function. Fig. 2(a) shows an example of $\Psi(.)$ which has been used later in this section to derive a variant of a non-spiking GT neuron model.

Rewriting (9) in terms of the objective function for the neuron model from (8), we get

$$v_{i,n+1} \leftarrow \frac{-g_i(v_n) + \lambda v_{i,n}}{-v_{i,n} g_i(v_n) + \lambda}, \quad i = 1, ..., M, \tag{13}$$

where

$$g_i(v_n) = \sum_{j=1}^{M} Q_{ij} v_{j,n} - b_{i,n} + \Psi(v_{i,n}). \tag{14}$$

Asymptotically from (12), and as shown in Appendix B, we have

$$\mathcal{E}_n \Big[(1 - v_{i,n} v_{i,n+1}) g_i(v_n) \Big] \to 0. \tag{15}$$

Since the energy functional in this case is continuously differentiable, the neural variables v_i converge to a local minimum, such that

$$\lim_{n \to \infty} v_{n+1} = v_n, \tag{16}$$

Therefore, (15) can be written as

$$(1 - v_{i,n}^2) g_i(v_n) \to 0. \tag{17}$$

As long as $v_i < 1$, $g_i(v_n) \to 0$, which from (14) leads to the following response for the i-th neuron

$$v_i = \Psi^{-1} \left(- \sum_{j=1}^{M} Q_{ij} v_j + b_i \right) \tag{18}$$

The nature and shape of $\int \Psi(v_i)dv_i$ determines the compressive response of the neuron given the pre-synaptic input and the input stimuli. For example, for the quadratic form of $\int \Psi(v_i)dv_i$ shown in Fig. 2(a), the corresponding derivative is $\Psi(v_i) = Cv_i$. When no external stimulus b_i is applied, the neuron response converges to $v_i = 0$, as shown in Fig. 2(b). When a positive stimulus

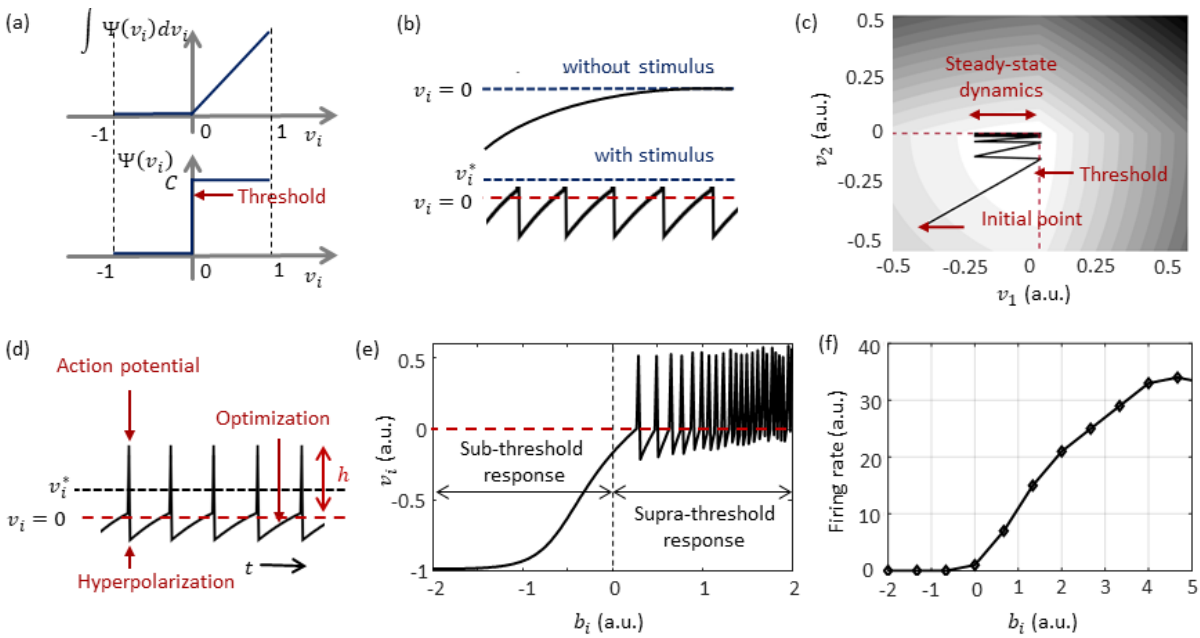

Figure 3: (a) The function $\int \Psi(v_i)dv_i$ and its derivative $\Psi(.)$ used in this paper for the spiking neuron model; (b) Time-evolution of the response v_i of a single neuron in the spiking model in the absence and presence of external stimulus; (c) Contour plot showing steady-state dynamics of the membrane potentials in a 2-neuron network in the presence of external stimulus; (d) The composite spike signal in the presence of external stimulus when the optimal solution shifts; (d) Plot of membrane potential v_i of the spiking neuron model vs external current stimulus; (e) Input-output characteristics for the spiking neuron model.

b_i is applied, the fixed-point shifts to a level that is a function of the stimulus magnitude, also shown in Fig. 2(b). This leads to an input-output characteristic shown in Fig. 2(c) for different values of the slope parameter C. Note that unlike conventional artificial neuron models, that use an explicit compressive non-linearity like a sigmoid or tanh function, the growth-transform dynamical system generates a compressive response as a first-order condition of an underlying optimization problem. In the next subsection we show that this framework could be extended to a spiking neuron model by introducing a gradient discontinuity in $\Psi(.)$.

2.2.2 Growth Transform spiking neuron model

The action potential or spike generated by a neuron is the result of a rapid change in its transmembrane current [2]. To model this effect, the transmembrane current in the compartmental model described in (8) can be approximated by a discontinuous $\Psi(.)$. In general, the regularization function $R(v_i) = \int \Psi(v_i)dv_i$ is chosen to be convex. We further select a form of $R(.)$ such that $R(v_i > 0) \geq 0$ and $R(0) = 0$. Fig. 3(a) shows one such form that has a gradient discontinuity at a threshold ($v_i = 0$) at which the neuron generates an action potential. The corresponding $\Psi(.)$, also shown in 3(a), is given by

$$\Psi(v_{i,n}) = \begin{cases} C & ; \quad v_{i,n} > 0 \\ 0 & ; \quad v_{i,n} \leq 0. \end{cases} \tag{19}$$

Here C is a hyper-parameter that determines the slope of $R(v_i)dv_i$, as shown in Fig. 3(a). When there is no external stimulus b_i, the neuron response converges to $v_i = 0$ as in the non-spiking case, as illustrated in Fig. 3(b). When a positive stimulus b_i is applied, the optimal solution for v_i, indicated by v_i^*, shifts upward to a level that is a function of the stimulus magnitude, also

shown in Fig. 3(b). However, due to the discontinuity in $\Psi(.)$, v_i^* is not directly reachable by the growth transform updates and v_i starts oscillating back and forth around the discontinuity (which, in our case, coincides with the threshold). Thus when $\Psi(.)$ is introduced, the potential v_i switches when $\Psi(v_i) > 0$ or only when $v_i > 0$. However, the dynamics of v_i remains unaffected for $v_i < 0$. The convergence process and the steady-state dynamics are illustrated in Fig. 3(c), which shows the contours of the energy function in (8) for a 2-neuron network. We assume that the neurons 1 and 2 are uncoupled, and only neuron 1 gets a positive stimulus. When the stimulus is applied, the updates push v_1 above the threshold, and in steady-state, v_1 shows a limit-cycle dynamics around the optimal solution.

Asymptotically, from (12), we get

$$\mathcal{E}_n[g(v_n)] = \mathcal{E}_n[v_n v_{n+1} g(v_n)]. \tag{20}$$

We can decompose $\alpha_n = v_n v_{n+1}$ into two terms as follows

$$\alpha_n = \alpha_0 + \Delta\alpha_n, \tag{21}$$

where α_0 is the temporal mean and $\Delta\alpha_n$ is a zero-mean fluctuation around α_0, i.e., $\mathcal{E}_n[\Delta\alpha_n] = 0$. From (20) we can write

$$(1 - \alpha_0)\mathcal{E}_n[g(v_n)] = \mathcal{E}_n[\Delta\alpha_n g(v_n)] \tag{22}$$

Under the assumption that $\Delta\alpha_n$ and $g(v_n)$ are uncorrelated, the R.H.S. of (22) is 0. This implies one of two conditions: (a) $\alpha_0 = 1$, which means both v_n and v_{n+1} are -1. This is the case when the neuron response has saturated to its lower bound. (b) $\mathcal{E}_n[g(v_n)] = 0$, which implies that asymptotically the network exhibits limit-cycles about a single attractor or a fixed-point such that the time-expectations of its state variables encode this optimal solution. A similar stochastic first-order framework was used in [25] to derive a dynamical system corresponding to $\Sigma\Delta$ modulation for tracking low-dimensional manifolds embedded in high-dimensional analog signal spaces.

For case (b), from (14) we have

$$\sum_{j=1}^{M} Q_{ij}\mathcal{E}_n[v_{j,n}] - \mathcal{E}_n[b_{i,n}] + \mathcal{E}_n[\Psi(v_{i,n})] = 0. \tag{23}$$

As before, we can show that (23) can be rewritten as

$$\mathcal{E}_n[v_{i,n}] = \sum_{j=1}^{M} W_{ij}\mathcal{E}_n[\Psi(v_{j,n})] + \mathcal{E}_n[y_i], \tag{24}$$

where W_{ij} is an element of the matrix $\boldsymbol{W}$ that satisfies $\boldsymbol{W} = -\boldsymbol{Q}^{-1}$. This ensures that for non-saturating neurons, (2) holds on an average over time.

During the brief period when $v_i > 0$, we assume that the neuron enters into a runaway state leading to a voltage spike. The composite spike signal $s_i(t)$, shown in Fig. 3(d), is then treated as a combination of the sub-threshold and supra-threshold responses and is given by

$$s_{i,n} = v_{i,n} + h\mathcal{I}[v_{i,n} > 0], \tag{25}$$

where the finite constant $h > 0$ determines the magnitude of the spike. Note that in the current version, the proposed model does not explicitly model the runaway process that leads to the spike, unlike other neuron models [2, 3, 4]. However, it does incorporate the hyperpolarization part of the spike as a result of v_i oscillating around the gradient discontinuity in order to encode the new optimal point, and hence a refractory period is automatically incorporated in between

two spikes. Because the switch at the time of an action potential happens for a single discrete time-step before the updates pull v_i back below threshold, a finite value of h ensures that

$$\mathcal{E}_n[v_{i,n}] = \mathcal{E}_n[s_{i,n}]. \tag{26}$$

Hence, all the results that hold for $\mathcal{E}_n[v_{i,n}]$ also hold for $\mathcal{E}_n[s_{i,n}]$.

$R(v_i) = \int \Psi(v_i)dv_i$ in the network energy functional in effect models the power dissipation due to spiking activity. For the form of $R(.)$ chosen in this paper, this power dissipation is taken to be zero below the threshold, and increases linearly above threshold. A plot of the membrane potential vs external current for the spiking neuron model is presented in Fig. 3(e). As v_i exceeds the threshold for a positive stimulus, the neuron enters a spiking regime and the firing rate changes almost linearly with the input, whereas the sub-threshold response is similar to the non-spiking case. Fig. 3(f) shows the tuning curve for the neuron as the input stimulus is increased. It is nearly sigmoidal in shape and shows how the firing rate reaches a saturation level for relatively high inputs. The proposed spiking neuron model based on the discrete-time Growth Transform dynamical system is summarized in Table 2.

Table 2: Discrete-time GT spiking neuron model

For a network of M neurons with state variables $\boldsymbol{v} = \{v_i\} \in \mathcal{R}^M$, where the synaptic weight matrix is denoted by $\boldsymbol{W} = \{W_{ij}\} \in \mathcal{R}^M \times \mathcal{R}^M$ and the external stimulus vector is denoted by $\boldsymbol{b} = \{b_i\} \in \mathcal{R}^M$, the time-evolution of the network under bound constraints on the state variables is governed by the following discrete-time updates:

$$v_{i,n+1} \leftarrow \frac{-g_i(v_n) + \lambda v_{i,n}}{-v_{i,n} g_i(v_n) + \lambda}, \quad i = 1, ..., M, \tag{27}$$

where

$g_i(v_n) = \sum_{j=1}^{M} Q_{ij} v_{j,n} - b_{i,n} + \Psi(v_{i,n})$;

Q_{ij} is an element of the transconductance coupling matrix $\boldsymbol{Q}$, where $\boldsymbol{Q} = -\boldsymbol{W}^{-1}$

$\Psi(v_{i,n}) = \left\{ \begin{array}{ccc} C & ; & v_{i,n} > 0 \\ 0 & ; & v_{i,n} \leq 0 \end{array} \right\}$;

λ is a fixed hyper-parameter such that $\lambda > |g_{i,n}| \ \forall i, n$.

The spike response of the i-th neuron at time t is given by

$$s_{i,n} = v_{i,n} + h\mathcal{I}[v_{i,n} > 0],$$

where h determines the magnitude of each spike.

2.3 Complete model of the Growth Transform neuron

Single neurons show a vast repertoire of response characteristics and dynamical properties that lend richness to their computational properties at the network level. Izhikevich in ([26]) provides an extensive review of different spiking neuron models and their ability to produce the different dynamics observed in biology. In Appendix C, we extend the proposed model into a continuous-time dynamical system, which enables us to reproduce a vast majority of such dynamics and also allows us to explore interesting properties in coupled networks. The complete neuron model is summarized in Table 3.

The operation of the proposed neuron model is therefore governed by two sets of dynamics: (a) minimization of the network energy functional $\mathcal{H}$; (b) modulation of the trajectory using the

parameter $f_{i,n}$ which can be time-varying and individually tuned for each neuron. Fortunately, the evolution of $f_{i,n}$ can be made as complex as possible without affecting the asymptotic fixed-point solution of the optimization process. It can be a made a function of local variables like v_i and $\dot{v}_i$ or a function of global/network variables like $\mathcal{H}$ and $\dot{\mathcal{H}}$. Different choices of the modulation function can lead to different trajectories followed by the neural variables under the same energy contour, as illustrated in Fig. 4. In the Results section, we show how different forms of $f_{i,n}$ produce different sets of neuronal dynamics consistent with the dynamics that have been reported in neurobiology.

Table 3: Complete continuous-time GT spiking neural network (Proof in Appendix C)

For a network of M neurons with state variables $\boldsymbol{v} = \{v_i\} \in \mathcal{R}^M$, where the synaptic weight matrix is denoted by $\boldsymbol{W} = \{W_{ij}\} \in \mathcal{R}^M \times \mathcal{R}^M$ and the external stimulus vector is denoted by $\boldsymbol{b} = \{b_i\} \in \mathcal{R}^M$, the time-evolution of the network under bound constraints on the state variables is governed by the following continuous-time dynamical system:

$$\frac{1}{f_i(t)}\frac{dv_i(t)}{dt} + v_i(t) = \frac{-g_i(t) + \lambda v_i(t)}{-v_i(t)g_i(t) + \lambda}, \tag{28}$$

where

$g_i(t) = \sum_{j=1}^{M} Q_{ij}v_j(t) - b_i(t) + \Psi(v_i(t))$;

Q_{ij} is an element of the transconductance coupling matrix $\boldsymbol{Q}$, where $\boldsymbol{Q} = -\boldsymbol{W}^{-1}$

$\Psi(v_i(t)) = \left\{ \begin{array}{ccc} C & ; & v_i(t) > 0 \\ 0 & ; & v_i(t) \leq 0 \end{array} \right\}$;

λ is a fixed hyper-parameter such that $\lambda > |g_i(t)| \ \forall i, t$;

$0 \leq f_i(t, v_i, \dot{v}_i, \mathcal{H}, \dot{\mathcal{H}}) \leq 1$ is a modulation function that can be tuned individually for each

neuron to encode different trajectories and different steady-state spiking dynamics corresponding

to the optimal solution.

The spike response of the i-th neuron at time t is given by

$$s_i(t) = v_i(t) + h\mathcal{I}[v_i(t) > 0], \tag{29}$$

where $h > 0$ is a finite number that determines the magnitude of each spike.

2.4 From neuron to network: Geometric interpretation of network dynamics

The remapping from standard coupled conditions of a spiking neural network to the proposed formulation admits an elegant geometric interpretation of neural dynamics. Similar to the network coding framework presented in [10], we show in this section how the activity of individual neurons in a network can be visualized with respect to a network hyper-plane. This geometric interpretation can then be used to understand network dynamics in response to different stimuli.

If we assume that the matrix $\boldsymbol{Q}$ is locally positive-definite, each of its elements Q_{ij} can be written as an inner product between two vectors as $Q_{ij} = \boldsymbol{x}_i.\boldsymbol{x}_j$, where $\boldsymbol{x}_i \in \mathcal{R}^D$ is a vector in a D-dimensional metric space. This is similar to kernel methods that compute similarity functions between pairs of vectors in the context of support vector machines [27]. This associates the i-th neuron in the network with a vector $\boldsymbol{x}_i$, mapping it onto an abstract metric space $\mathcal{R}^D$ and essentially providing an alternate geometric representation of the neural network. From (23),

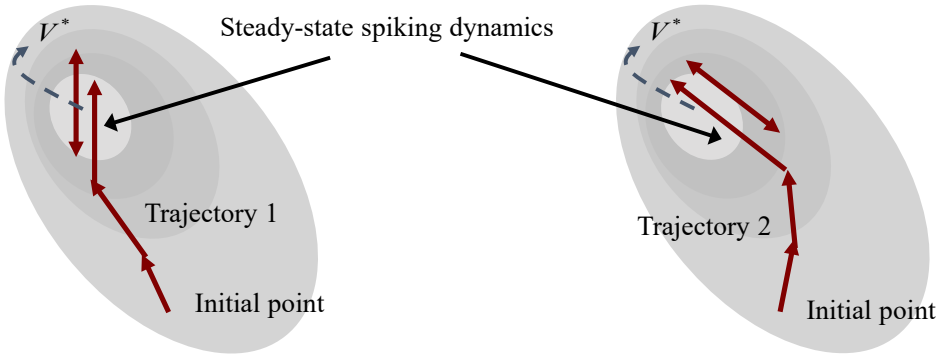

Figure 4: Different modulation functions lead to different steady-state spiking dynamics under the same energy contour.

the spiking activity of the i-th neuron can then be represented as

$$\begin{aligned} \mathcal{E}_n[\Psi(v_{i,n})] &= -\sum_{j=1}^{M} Q_{ij}\mathcal{E}_n[v_{j,n}] + \mathcal{E}_n[b_{i,n}] \\ &= \sum_{j=1}^{M} -(\boldsymbol{x}_i.\boldsymbol{x}_j)\mathcal{E}_n[v_{j,n}] + \mathcal{E}_n[b_{i,n}] \\ &= \boldsymbol{w}.\boldsymbol{x}_i + \bar{b}_i, \end{aligned} \tag{30}$$

where

$$\boldsymbol{w} = -\sum_{j=1}^{M} \boldsymbol{x}_j\mathcal{E}_n[v_{j,n}]. \tag{31}$$

$\mathcal{E}_n[\Psi(.)]$ therefore represents the distance of the vector $\boldsymbol{x}_i$ from a network hyperplane in the D-dimensional metric space, which is parameterized by the weight vector $\boldsymbol{w}$ and offset $\bar{b}_i$. When a stimulus $\bar{b}_i$ is applied, the hyperplane shifts, leading to a stimulus-specific value of this distance for each neuron that is also dependent on the network configuration Q. Hence, $\Psi(.)$ is denoted as a 'network variable', that signifies how the response of each neuron is connected to the rest of the network. Note that we can also write the synaptic weight elements in a kernelized form as $Q_{ij} = K(\boldsymbol{x}_i).K(\boldsymbol{x}_j)$, where $K(.)$ is a nonlinear transformation function, defining a non-linear boundary for each neuron. This idea of a dynamic and stimulus-specific hyperplane can offer intuitive interpretations about several population dynamics reported in literature and have been elaborated on in the Results section.

3 Results

3.1 Single-neuron dynamics

We first show how we can reproduce a number of single-neuron response characteristics by changing the modulation function $f_i(t)$ in the neuron model. For this, we consider an uncoupled network, where

$$Q_{ij} = \begin{cases} 1 & ; \quad \forall i = j \\ 0 & ; \quad \forall i \neq j \end{cases}. \tag{32}$$

We will subsequently extend these dynamics to build coupled networks with interesting properties like memory and global adaptation for energy-efficient neural representation. The results reported here are representative of the types of dynamical properties the proposed model can exhibit,

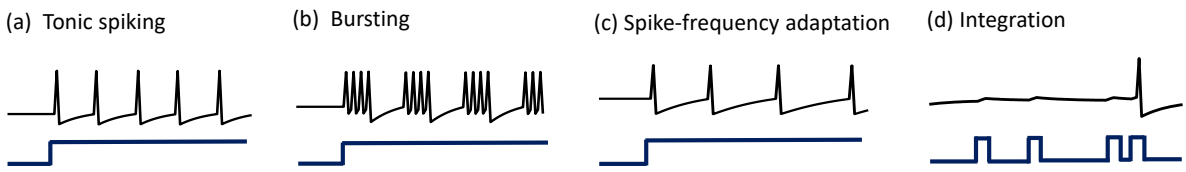

Figure 5: (a)-(d) Simulations demonstrating different single-neuron responses obtained using the GT neuron model.

but are by no means exhaustive. Readers are encouraged to experiment with different inputs and network parameters in the software (MATLAB©) implementation of the Growth Transform neuron model [11]. The tool enables users to visualize the effects of different modulation functions and other hyperparameters on the neural dynamics, as well as the time-evolution of population trajectories and the network energy function with different inputs and under different initial conditions.

3.1.1 Standard tonic-spiking response

When stimulated with a constant current stimulus b_i, a vast majority of neurons fire single, repetitive action potentials for the duration of the stimulus, with or without adaptation [28, 29, 30]. The proposed model shows tonic spiking without adaptation when the modulation function $f_{i,n} = f$, where $0 < f \leq 1$. A simulation of tonic spiking response using the neuron model is given in Fig. 5(a).

3.1.2 Bursting response

Bursting neurons fire discrete groups of spikes interspersed with periods of silence in response to a constant stimulus [28, 29, 31, 32]. Bursting arises from an interplay of fast ionic currents responsible for spiking, and slower intrinsic membrane currents that modulate the spiking activity, causing the neuron to alternate between activity and quiescence. Bursting response can be simulated in the proposed model by modulating $f_{i,n}$ at a slower rate compared to the generation of action potentials, in the following way

$$f_{i,n} = \begin{cases} f & ; \quad c_{i,n} < B \\ 1 & ; \quad c_{i,n} \geq B \end{cases} \tag{33}$$

where B is a hyper-parameter and the count variable $c_i(n)$ is updated according to

$$c_{i,n} = \begin{cases} c_{i,n-1} + \Psi(v_{i,n}) & ; \quad c_{i,n-1} < B \\ 0 & ; \quad c_{i,n-1} \geq B \end{cases} \tag{34}$$

Simulation of a bursting neuron in response to a step input is given in Fig. 5(b).

3.1.3 Spike-frequency adaptation

When presented with a prolonged stimulus of constant amplitude, many cortical cells initially respond with a high-frequency spiking that decays to a lower steady-state frequency [33]. This adaptation in the firing rate is caused by a negative feedback to the cell excitability due to the gradual inactivation of depolarizing currents or activation of slow hyperpolarizing currents upon depolarization, and occur at a time-scale slower than the rate of action potential generation. We

modeled spike-frequency adaptation by varying the modulation function according to

$$f_{i,n} = \begin{cases} \sigma[-h_n * \Psi(v_{i,n})] & ; \quad v_{i,n} < 0 \\ 1 & ; \quad v_{i,n} \geq 0, \end{cases} \tag{35}$$

where $h_n * \Psi(v_{i,n})$ is a convolution operation between the time-domain filter h_n and the spiking function $\Psi(v_{i,n})$, and

$$\sigma(x) = b\left[1 + \frac{1}{1 + \exp(-x)}\right] \tag{36}$$

is a sigmoidal function ensuring $1 > f_{i,n} > 0$. The parameter $b \in (0, 1)$ determines the steady-state firing rate for a particular stimulus. For the results, the smoothing filter h_n has been chosen to be

$$H(z) = \frac{1}{1 - 0.99z^{-1} - 0.01\log_2(1+\beta)z^{-1}}, \tag{37}$$

where $\beta > 0$ is a hyper-parameter for tuning the first-order filter response. As the neuron spikes, f decreases, slowing down the growth-transform neuron updates until the frequency reaches a lower constant value. A tonic-spiking response with spike-frequency adaptation is shown in Fig. 5(c).

3.1.4 Integrator response

When the baseline input is set slightly negative so that the fixed point is below the threshold, the neuron works like a leaky integrator as shown in Fig. 5(d), preferentially spiking to high-frequency or closely-spaced input pulses that are more likely to make v_i cross the threshold.

3.2 Coupled spiking network with pre-synaptic adaptation

We can extend the proposed framework to a network model where the neurons, apart from external stimuli, receive inputs from other neurons in the network. We begin by considering $\mathbf{Q}$ to be a positive-definite matrix, which gives an unique solution of (8). Interactions among the neurons in a coupled network can be captured by the modulation function as follows

$$f_{i,n} = \begin{cases} \sigma[h_n * \sum_j Q_{ij}\Psi(v_{j,n})] & ; \quad v_{i,n} < 0 \\ 1 & ; \quad v_{i,n} \geq 0 \end{cases}, \tag{38}$$

with the compressive-function $\sigma(.)$ and the smoothing-filter $h(n)$ given by (36) and (37). This ensures that $Q_{ij} > 0$ corresponds to an excitatory coupling from the pre-synaptic neuron j, and $Q_{ij} < 0$ corresponds to an inhibitory coupling, as demonstrated in Fig. 6(a). Fig. 6(b) illustrates how different variations of the modulation function under the same energy landscape leads to the neurons following different response trajectories and converging to varying limit-cycles. The energy contours for a two-neuron network corresponding to a particular $\boldsymbol{Q}$ with excitatory and inhibitory connections and a fixed stimulus vector $\boldsymbol{b}$ are plotted, along with the responses of the two neurons starting from the same initial conditions, with and without pre-synaptic adaptation (where the latter corresponds to the case where neither neuron can integrate individual spike inputs from the other neuron and the only coupling is through $\boldsymbol{Q}$). Because the energy landscape is the same in both cases, the neurons converge to the same sub-domain, but with widely varying trajectories and steady-state response patterns.

3.3 Coupled network with pre-synaptic and global adaptation

Apart from the pre-synaptic adaptation that changes individual firing rates based on the inputs received by each neuron, neurons in the coupled network can be made to adapt according to the

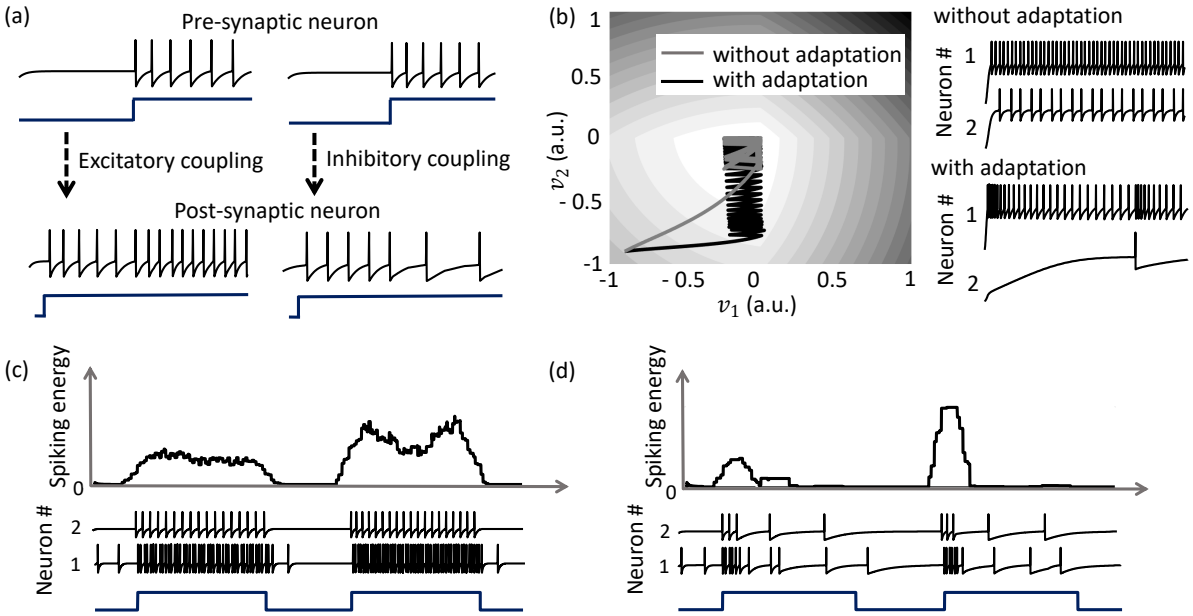

Figure 6: (a) Results from a 2-neuron network with excitatory and inhibitory couplings; (b) Energy optimization process under different conditions lead to different limit cycles within the same energy landscape. (c)-(d) Mean spiking energy $\int \Psi(v_i)dv_i$ and firing patterns in response to a step input without and with global adaptation respectively.

global dynamics by changing the modulation function as follows

$$f_{i,n} = \begin{cases} \sigma[(h_n * \sum_j Q_{ij}\Psi(v_{j,n})) + \mathcal{F}(\mathcal{H}, \dot{\mathcal{H}})] & ; \quad v_{i,n} < 0 \\ 1 & ; \quad v_{i,n} \geq 0 \end{cases} \tag{39}$$

with the compressive-function $\sigma(.)$ and smoothing-filter h_n given by (36) and (37). The new function $\mathcal{F}(.)$ is used to capture the dynamics of the network cost-function. As the network starts to stabilize and converge to a fixed-point, the function $f_i(.)$ adapts to reduce the spiking rate of the neuron without affecting the steady-state solution. Fig.s 6(c) and (d) shows the time-evolution of the spiking energy $\int \Psi(.)dv$ and the spike-trains for a two-neuron network without global adaptation and with global adaptation respectively, using the following form for the adaptation term

$$\mathcal{F}(\mathcal{H}, \dot{\mathcal{H}}) = \begin{cases} -\sigma[h_n * \mathcal{E}_t(\mathcal{H})] & ; \quad \mathcal{E}_t(\dot{\mathcal{H}}) \approx 0 \\ 0 & ; \quad \text{otherwise.} \end{cases} \tag{40}$$

This feature is important in designing energy-efficient spiking networks where energy is only dissipated during transients.

3.4 Network response and network trajectories

In order to outline the premises of the next few experiments on population dynamics using this intuition, we consider a small network of neurons on a two-dimensional co-ordinate space, and assign arbitrary inputs to the neurons. A Gaussian kernel is chosen for the coupling matrix Q as follows

$$Q_{ij} = \exp(-\gamma||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2). \tag{41}$$

This essentially clusters neurons with stronger couplings between them closer to each other on the metric space, while placing neurons with weaker couplings far away from each other.

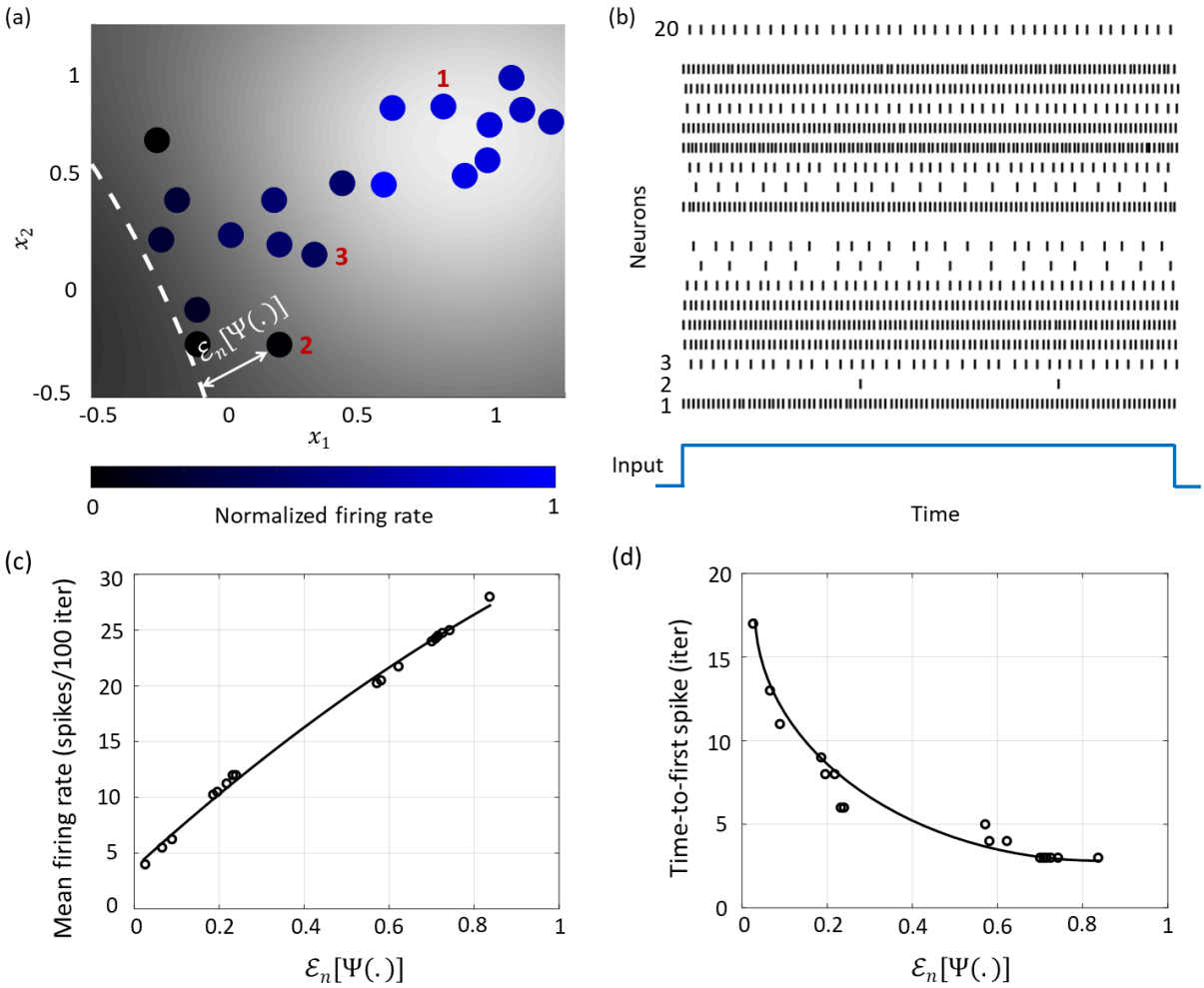

Figure 7: (a) Contour plot of spiking activity corresponding to a particular stimulus vector. Neurons are colored according to their mean firing rate (normalized w.r.t. the maximum firing rate) during the stimulus period. The white dashed line is the contour corresponding to $\mathcal{E}_n[\Psi(.)] = 0$. (b) Spike raster for all neurons for the input in (a). (c) The mean firing rate and (d) time-to-first spike as a function of the distance for the network.

A network consisting of 20 neurons is shown in Fig. 7(a), which also shows how the spiking activity changes as a function of the location for the particular network configuration and input stimulus vector. Each neuron is color coded based on the mean firing rate (normalized w.r.t. the maximum mean firing rate) with which it responds when the stimulus is on. Fig. 7(b) shows the spike raster for the entire network.We see that the responsiveness of the neurons to a particular stimulus is correlated with the distance at which it is located from the hypothetical hyperplane in the high-dimensional space to which the neurons are mapped through kernel transformation. We show below how this geometric representation can provide insights on population-level dynamics in the network considered.

3.4.1 Rate and temporal coding

The growth transform neural network inherently shows a number of encoding properties that are commonly observed in biological neural networks [13, 34]. For example, the firing rate averaged over a time window is a popular rate coding technique that claims that the spiking frequency or rate increases with stimulus intensity [35]. A temporal code like the time-to-first-spike posits that a stronger stimulus brings a neuron to the threshold faster, generating a spike, and hence

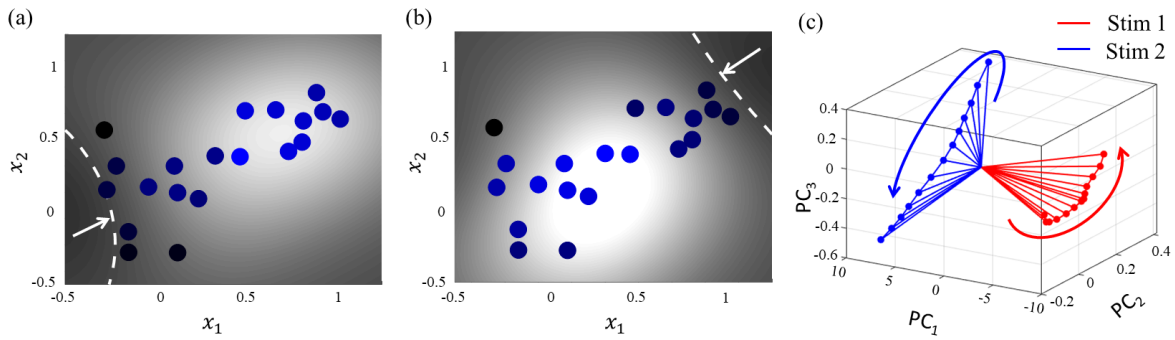

Figure 8: (a) and (b) Perturbation of the stimulus vector in different directions for the same network produces two different contours. (c) Corresponding population activities trace different trajectories in the neural subspace.

relative spike arrival times contain critical information about the stimulus [36].

These coding schemes can be interpreted under the umbrella of network coding using the same geometric representation as considered above. Here, the responsiveness of a neuron is closely related to its proximity to the boundary (or the contour line corresponding to z = 0). The effective stimulus strength is more for neurons which exhibit more spiking, and have a higher value of the distance from the boundary. We see from Fig.s 7(c) and (d) that as this value increases, the average firing rate of a neuron (number of spikes in a fixed number of iterations) increases, and the time-to-first spike becomes progressively smaller. Neurons with a distance value below a certain threshold do not spike at all during the stimulus period, and therefore have a mean firing rate of zero and time-to-spike at infinity. Therefore, based on how the network is configured in terms of synaptic inputs and connection strengths, the spiking pattern of individual neurons conveys critical information about the boundary and their placement with respect to it.

3.4.2 Network coding and neural population trajectories

The encoding of a stimulus in the spatiotemporal evolution of activity in a large population of neurons is often represented by an unique trajectory in a high-dimensional space, where each dimension accounts for the time-binned spiking activity of a single neuron. Projection of the high-dimensional activity to two or three critical dimensions using dimensionality reduction techniques like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) have been widely used across organisms and brain regions to shed light on how neural population response evolves when a stimulus is delivered. For example in identity coding, trajectories corresponding to different stimuli evolve towards different regions in the reduced neural subspace, that often become more discriminable with time and are stable over repeated presentations of a particular stimulus [37, 38, 39]. We show how this can be explained in the context of the geometric interpretation.

For the same network as above, we start with the simplest possible experiment, starting from the same baseline, and perturbing the stimulus vector in two different directions. This pushes the boundary in two different directions, exciting different subsets of neurons, as illustrated in Fig.s 8(a) and (b). A similar dimensionality reduction to three principal components in Fig. 8(c) shows the neural activity unfolding in distinct stimulus-specific areas of the neural subspace. The two contour plots also show that some neurons may spike for both the inputs, while some spike selectively for one of them. Yet others may not show any spiking for either stimulus, but may spike for some other stimulus vector and the corresponding stimulus-specific boundary.

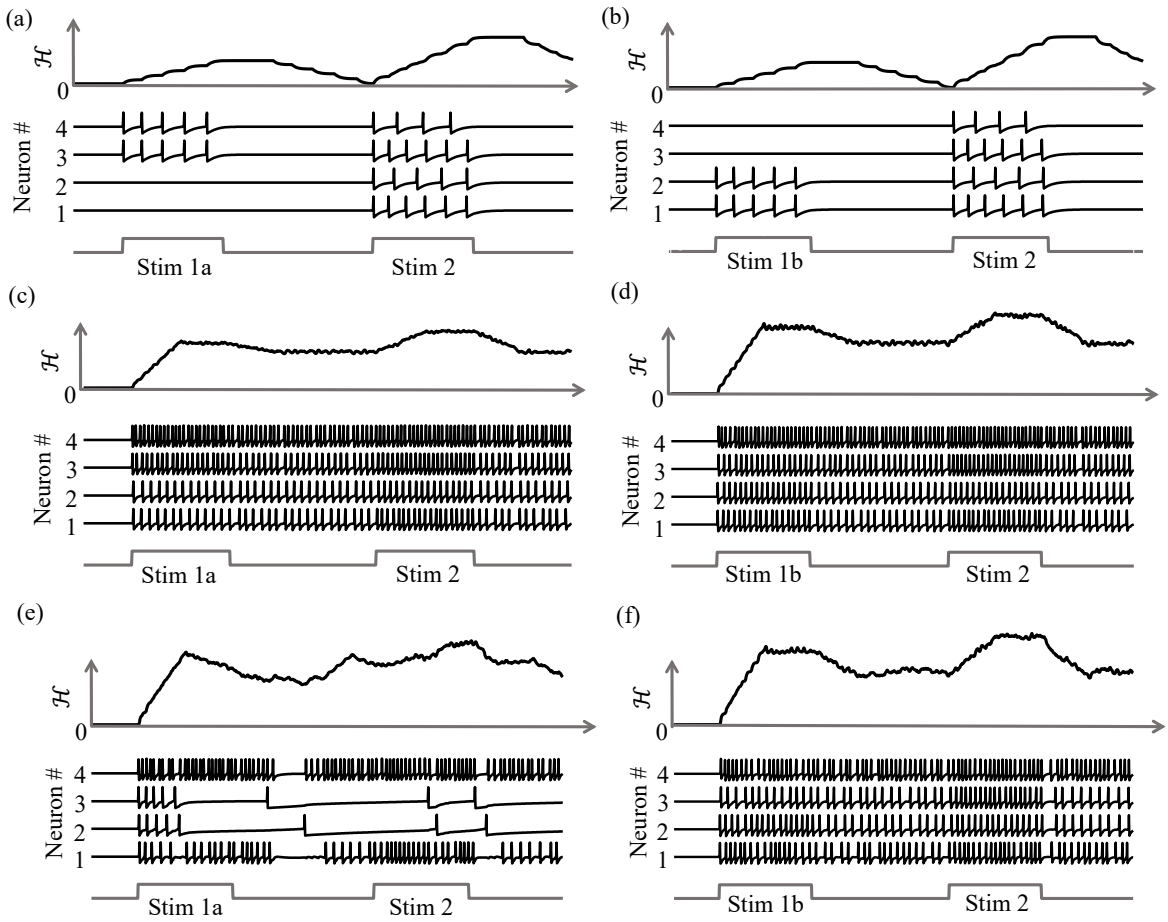

Figure 9: Stimulus response for a 4-neuron network with different stimulus histories for: (a)-(b) an uncoupled network; (c)-(d) a coupled network with a positive definite synaptic weight matrix; (e)-(f) a coupled network with a non-positive definite synaptic weight matrix.

3.5 Coupled spiking network with non-positive definite Q

As illustrated in Fig. 9, a coupled spiking network can function as a memory element, when $\mathbf{Q}$ is a non-positive definite matrix and

$$f_{i,n} = \begin{cases} \sigma[h_n * \sum_j Q_{ij}\Psi(v_{j,n})] & ; \quad v_{i,n} < 0 \\ 1 & ; \quad v_{i,n} \geq 0 \end{cases} \tag{42}$$

due to the presence of more than one attractor state. We demonstrate this by considering two different stimulus histories in a network of four neurons, where a stimulus 'Stim 1a' precedes another stimulus 'Stim 2' in Fig.s 9(a), (c) and (e), and a different stimulus 'Stim 1b' precedes 'Stim 2' in Fig.s 9(b), (d) and (f). Here, each 'stimulus' essentially corresponds to a different input vector $\boldsymbol{b}$. For an uncoupled network, where neurons do not receive any inputs from other neurons, the network energy increases when the first stimulus is applied and returns to zero afterwards, and the network begins from the same state again for the second stimulus as for the first, leading to the same firing pattern for the second stimulus, as shown in Fig.s 9(a) and (b), independent of the history. For a coupled network with a positive definite synaptic weight matrix Q, reinforcing loops of spiking activity in the network may not allow the network energy to go to zero after the first stimulus is removed, and the residual energy may cause the network to exhibit a baseline activity that depends on stimulus history, as long as there is no dissipation. When the second stimulus is applied, the initial conditions for the network are different for the two stimulus histories, leading to two different transients until the network settles down into the same steady-state firing patterns, as shown in Fig.s 9(c) and (d). For a non-positive definite synaptic weight matrix Q however, depending on the initial condition, the network may settle down to different solutions for the same second stimulus, due to the possible presence of more than one local minimum. This leads to completely different transients as well as steady-state responses for the second stimulus, as shown in Fig. 9(e) and (f). This history-dependent stimulus response could serve as a short-term memory, where residual network energy from a previous external input subserves synaptic interactions among a population of neurons to set specific initial conditions for a future stimulus based on the stimulus history, forcing the network to settle down in a particular attractor state.

3.6 Associative memory network using Growth Transform neuron models

Associative memories are neural networks which can store memory patterns in the activity of neurons in a network through a Hebbian modification of their synaptic weights; and recall a stored pattern when stimulated with a partial fragment or a noisy version of the pattern [40]. Various works have studied associative memories using networks of spiking neuron models having different degrees of abstraction and architectural complexities [41, 42]. Here, we demonstrate using an associative memory network using Growth Transform neurons how we can use network trajectories to recall stored patterns, and moreover, use global adaptation to do so using very few spikes and high recall accuracy.

Our network comprises $M = 100$ neurons, out of which a randomly selected subset $m = 10$ are active for any stored memory pattern. The elements of the transconductance coupling matrix are set according to the following standard Hebbian learning rule

$$Q_{ij} = \frac{1}{M}\sum_{p=1}^{P} t_i^p t_j^p, \tag{43}$$

where $\boldsymbol{t}^p \in [0,1]^M,\ p = 1, ..., P$, are the binary patterns stored in the network. During the recall phase, only half of the cells active in the original memory are stimulated with a steady depolarizing input, and the spiking pattern across the network is recorded. Instead of determining

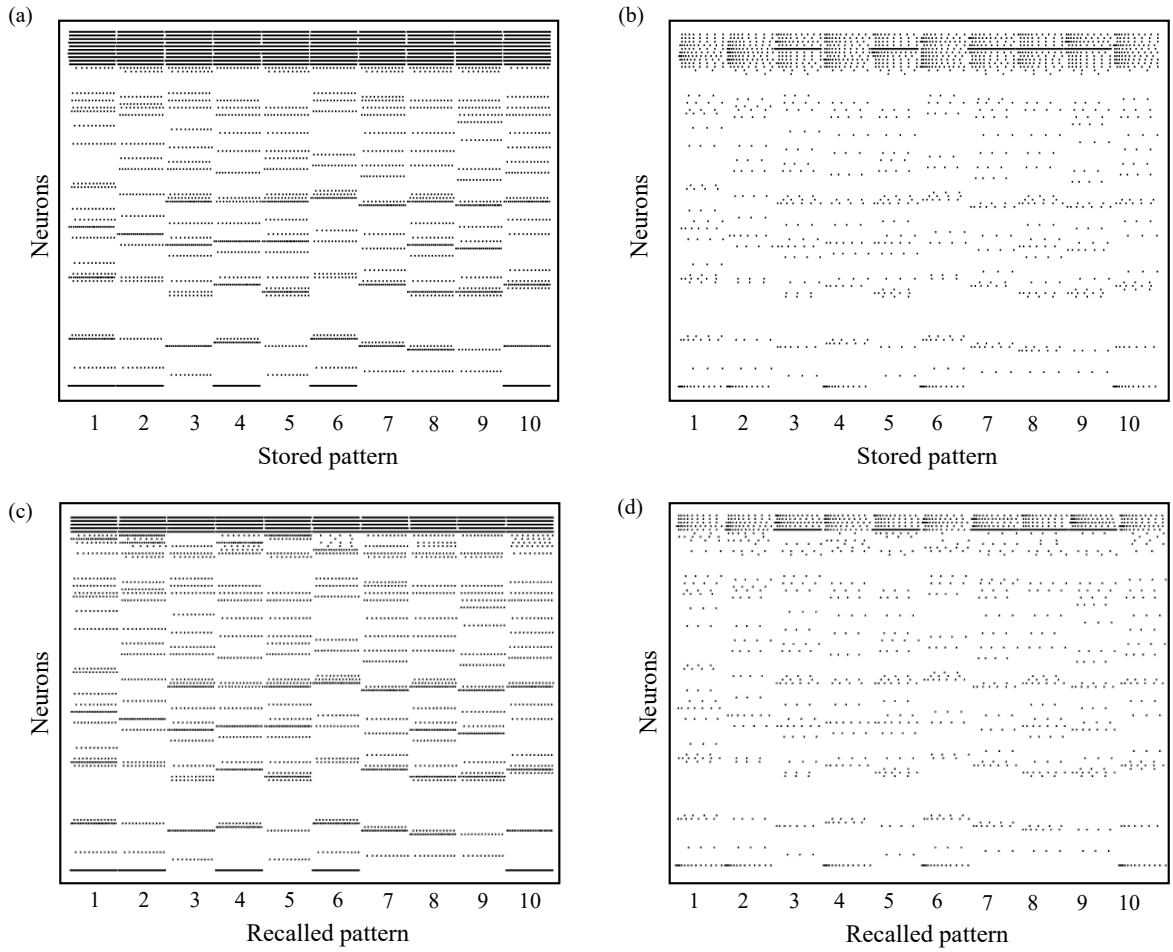

Figure 10: (a)-(b) Spike rasters for the 10 stored patterns without and with global adaptation respectively; (c)-(d) Spike rasters for the 10 recall cases without and with global adaptation respectively.

the active neurons during recall through thresholding and directly comparing with the stored binary pattern, we quantitatively measure the recall performance of the network by computing the mean distance between each pair of original-recall spiking dynamics as they unfold over time. This ensures that we not only take into account the firing of the neurons that belong to the pattern albeit are not directly stimulated, but also enables us to exploit any contributions from the rest of the neurons in making the spiking dynamics more dissimilar in comparison to recalls for other patterns.

When the network is made to globally adapt according to the system dynamics, the steady-state trajectories can be encoded using very few spikes. Fig.s 10 (a) and (b) show the raster plots for the stored patterns without and with global adaptation respectively when $P = 10$; and Fig.s 10 (c) and (d) are the corresponding plots during recall. For each recall pattern, spike patterns for the directly stimulated neurons are plotted first, followed by the other 5 neurons that are not directly stimulated but belong to the pattern; and finally the rest of the neurons in random order. The ordering of neurons is kept the same for plotting spike rasters for the stored patterns. During decoding, a straightforward metric using the average distance between time-binned mean firing rates for the original and recall trajectories produces similarity matrices presented in Fig.s 11(a) and (b), where we see that global adaptation does not perform as well. However, the information in this case also lies in the spike-times and changes in firing rate over time for each neuron. Including these features in the decoding vectors for stored and recalled

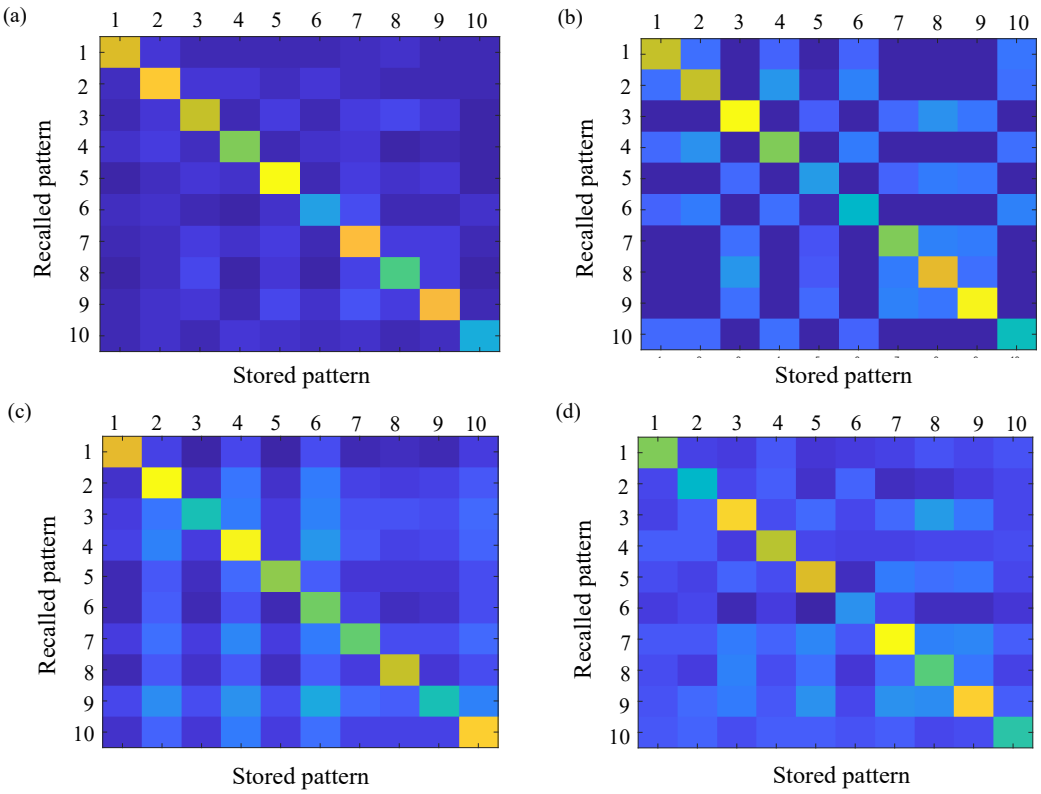

Figure 11: (a)-(b) Similarity matrices between storage and recall with a rate-based decoding metric; (c)-(d) Similarity matrices with a decoding metric that also includes spike-times and changes in mean firing rate.

patterns, we get clean recalls in both cases as shown in Fig.s 11(c) and (d). The decoding vector for the n-th time-bin in this case is given by

$$\boldsymbol{d}_n = \begin{bmatrix} \boldsymbol{r}_n \\ \Delta \boldsymbol{t}_{\Psi n} \\ \Delta \boldsymbol{r}_n \end{bmatrix}, \tag{44}$$

where $\boldsymbol{r}_n$, $\Delta \boldsymbol{t}_{\Psi n}$ and $\Delta \boldsymbol{r}_n$ are the mean firing rate, mean inter-spike interval and the change in mean firing rate for the n-th bin over the entire network respectively. Note that since the inter-spike interval is computed only for the time bin under consideration, it may be different from $(1/r_n)$, particularly for low firing rates, and hence encodes useful information. It is set equal to the bin length if there is a single spike over the entire bin length, and equal to twice the bin length if there are none. The similarity metric between the p-th stored pattern and the q-th recall pattern is given by

$$s_{p,q} = 1/\text{dist}_{p,q}; \tag{45}$$

where

$$\text{dist}_{p,q} = \frac{1}{N_{bins}} \sum_{n=1}^{N_{bins}} ||\boldsymbol{d}_{n,p_{\text{stored}}} - \boldsymbol{d}_{n,q_{\text{recall}}}||_2, \tag{46}$$

N_{bins} being the total number of time-bins over which each pattern was presented.

To estimate the capacity of the network, we calculate the mean recall accuracy over 10 trials for varying number of stored patterns. Fig. 12(a) plots the mean recall accuracy for different number of patterns stored for the two cases, and Fig. 12(b) plots the mean number of spikes for each storage. As expected, the accuracy is 100% for lesser storage, but degrades with higher

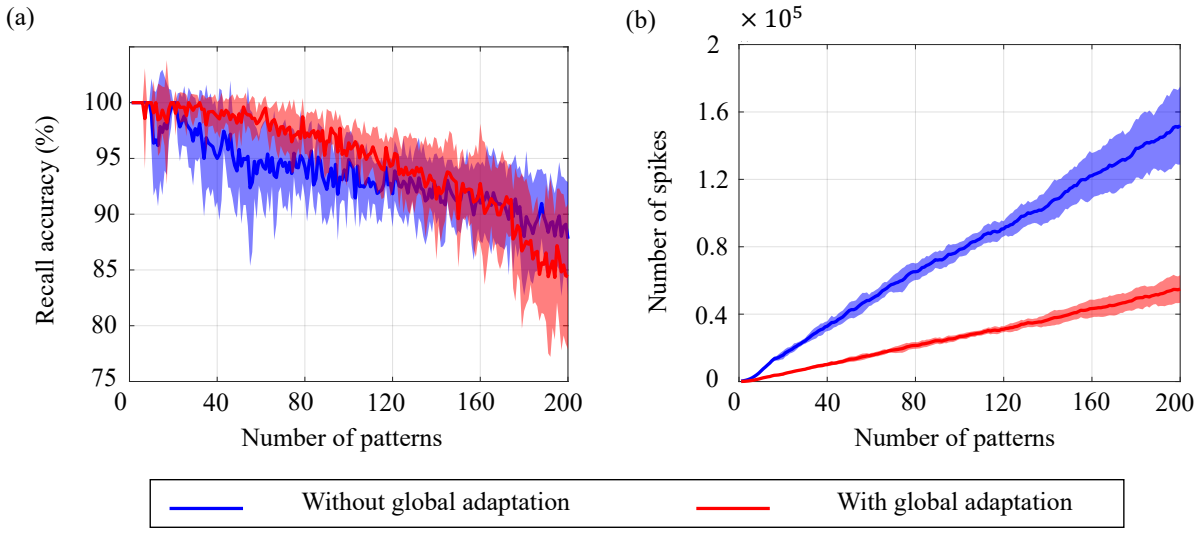

Figure 12: (a) Mean recall accuracy and (b) mean number of spikes as memory load increases for the network, in the absence as well as presence of global adaptation.

loading. However with global adaptation, the degradation is seen to be more graceful for a large range of storage with the decoding used in Fig.s 11(c) and (d), allowing the network to recall patterns more accurately using much fewer spikes. Hence by exploiting suitable decoding techniques, we can implement highly energy-efficient spiking associative memory networks with high storage capacity.

4 Conclusions

This paper introduces the theory behind a new spiking neuron and population model based on the Growth transform dynamical system. The system minimizes an appropriate energy functional under realistic physical constraints to produce emergent spiking activity in a population of neurons. The neuron model and its response are tightly coupled to the network objective, and is flexible enough to incorporate different neural dynamics that have been observed at the cellular level in electrophysiological recordings. In each of these experiments, the spiking activity is modulated externally by allowing a slower modulation function to evolve in different ways, without affecting the underlying optimization process. This allows the network to reach the same asymptotic fixed-point through many possible response trajectories, and using widely varying encoding techniques. The model can be easily extended to a coupled network of neurons, which can be tweaked to exhibit useful properties like short-term memory and global adaptation for energy-efficiency.

The paper is accompanied by a software tool [11] that enables readers to visualize the effects of different model parameters on the neural dynamics. Many more neural dynamics can be simulated using the model and readers are encouraged to experiment with different network hyper-parameters. The paper and the tool illustrate how spike generation and spike-based encoding in an artificial neuron model can be framed as a network-level energy optimization problem, thereby providing a framework to develop scalable and energy-efficient neuromorphic machine learning algorithms.

Although the current paper does not consider adaptation of synaptic weights, the general approach offers an elegant way to design neuromorphic machine learning algorithms. Energy-based frameworks are the backbone of most machine learning models, which perform learning and inference by associating a scalar value with each network state, denoting the 'energy' of the state, that captures the dependencies between its variables [12]. The non-differentiable

nature of spiking dynamics makes it hard to design energy-based learning models involving neural variables. Neuromorphic algorithms currently work around this problem in different ways, including mapping deep neural nets to spiking networks through rate-based techniques [43, 44], formulating loss functions that penalize the difference between actual and desired spike-times [45, 46], or approximating the derivatives of spike signals through various means [47, 48, 49]. However, formulating the spiking dynamics of the entire network using an energy function involving neural state variables across the network would enable us to directly use the energy function itself for learning weight parameters; and forms the basis for our future work. Since the proposed energy function encompasses all the neurons in the network, and not just the 'visible neurons' as in most neuromorphic machine learning algorithms, it can potentially enable easier and more effective training of hidden neurons in deep networks.

Conflict of Interest Statement

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Author Contributions

AG and SC contributed to the conception and design of the study; AG and DM conducted the simulations; AG, DM and SC designed the MATLAB interface; AG and SC wrote the first draft of the manuscript. All authors contributed to the manuscript revision, read and approved the submitted version.

Funding

This work was supported in part by a research grant from the National Science Foundation (ECCS: 1935073). D.M. was supported by a research grant from the National Institutes of Health (5R21EY02836202).

Acknowledgments

The authors would like to thank Dr. Kenji Aono at the Electrical and Systems Engineering department, Washington University, for developing a GPU version of the GT neural network model which is also included with the accompanying software toolbox [11].

References

[1] E. M. Izhikevich, *Dynamical systems in neuroscience.* MIT press, 2007.

[2] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of physiology*, vol. 117, no. 4, pp. 500–544, 1952.

[3] R. FitzHugh, "Impulses and physiological states in theoretical models of nerve membrane," *Biophysical journal*, vol. 1, no. 6, pp. 445–466, 1961.

[4] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on neural networks*, vol. 14, no. 6, pp. 1569–1572, 2003.

[5] L. F. Abbott, "Lapicques introduction of the integrate-and-fire model neuron (1907)," *Brain research bulletin*, vol. 50, no. 5-6, pp. 303–304, 1999.

[6] Z. Jonke, S. Habenschuss, and W. Maass, "Solving constraint satisfaction problems with networks of spiking neurons," *Frontiers in neuroscience*, vol. 10, p. 118, 2016.

[7] T. Nakano, M. Otsuka, J. Yoshimoto, and K. Doya, "A spiking neural network model of model-free reinforcement learning with high-dimensional sensory input and perceptual ambiguity," *PloS one*, vol. 10, no. 3, p. e0115620, 2015.

[8] E. P. Frady and F. T. Sommer, "Robust computation with rhythmic spike patterns," *arXiv preprint arXiv:1901.07718*, 2019.

[9] J. Burroni, P. Taylor, C. Corey, T. Vachnadze, and H. T. Siegelmann, "Energetic constraints produce self-sustained oscillatory dynamics in neuronal networks," *Frontiers in neuroscience*, vol. 11, p. 80, 2017.

[10] A. Gangopadhyay and S. Chakrabartty, "Spiking, bursting, and population dynamics in a network of growth transform neurons," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2379–2391, 2018.

[11] D. Mehta, A. Gangopadhyay, K. Aono, and S. Chakrabartty, "Growth transform neuron model matlab gui," https://github.com/aimlab-wustl/growth-transform-NN, 2019.

[12] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, "A tutorial on energy-based learning," *Predicting structured data*, vol. 1, no. 0, 2006.

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

[14] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[15] A. S. Cassidy, P. Merolla, J. V. Arthur, S. K. Esser, B. Jackson, R. Alvarez-Icaza, P. Datta, J. Sawada, T. M. Wong, V. Feldman *et al.*, "Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2013, pp. 1–10.

[16] R. D. Traub and R. Miles, *Neuronal networks of the hippocampus*. Cambridge University Press, 1991, vol. 777.

[17] D. Mar, C. Chow, W. Gerstner, R. Adams, and J. Collins, "Noise shaping in populations of coupled model neurons," *Proceedings of the National Academy of Sciences*, vol. 96, no. 18, pp. 10 450–10 455, 1999.

[18] S. H. Wright, "Generation of resting membrane potential," *Advances in physiology education*, vol. 28, no. 4, pp. 139–142, 2004.

[19] C. Koch and T. Poggio, "Biophysics of computation: neurons, synapses and membranes," 1984.

[20] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[21] G. E. Hinton, T. J. Sejnowski *et al.*, "Learning and relearning in boltzmann machines," *Parallel distributed processing: Explorations in the microstructure of cognition*, vol. 1, no. 282-317, p. 2, 1986.

[22] E. Gardner and B. Derrida, "Optimal storage properties of neural network models," *Journal of Physics A: Mathematical and general*, vol. 21, no. 1, p. 271, 1988.

[23] L. E. Baum and G. Sell, "Growth transformations for functions on manifolds," *Pacific Journal of Mathematics*, vol. 27, no. 2, pp. 211–227, 1968.

[24] A. Gangopadhyay, O. Chatterjee, and S. Chakrabartty, "Extended polynomial growth transforms for design and training of generalized support vector machines," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 5, pp. 1961–1974, 2017.

[25] A. Gore and S. Chakrabartty, "A min-max optimization framework for designing sigma-delta learners: Theory and hardware," *IEEE Trans. Circuits Syst. I*, vol. 57, no. 3, pp. 604–617, 2010.

[26] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE transactions on neural networks*, vol. 15, no. 5, pp. 1063–1070, 2004.

[27] S. Chakrabartty and G. Cauwenberghs, "Gini support vector machine: Quadratic entropy based robust multi-class probability regression," *Journal of Machine Learning Research*, vol. 8, no. Apr, pp. 813–839, 2007.

[28] A. Agmon and B. Connors, "Repetitive burst-firing neurons in the deep layers of mouse somatosensory cortex," *Neuroscience letters*, vol. 99, no. 1-2, pp. 137–141, 1989.

[29] D. A. McCormick, B. W. Connors, J. W. Lighthall, and D. A. Prince, "Comparative electrophysiology of pyramidal and sparsely spiny stellate neurons of the neocortex," *Journal of neurophysiology*, vol. 54, no. 4, pp. 782–806, 1985.

[30] J. R. Gibson, M. Beierlein, and B. W. Connors, "Two networks of electrically coupled inhibitory neurons in neocortex," *Nature*, vol. 402, no. 6757, p. 75, 1999.

[31] C. M. Gray and D. A. McCormick, "Chattering cells: superficial pyramidal neurons contributing to the generation of synchronous oscillations in the visual cortex," *Science*, vol. 274, no. 5284, pp. 109–113, 1996.

[32] J. C. Brumberg, L. G. Nowak, and D. A. McCormick, "Ionic mechanisms underlying repetitive high-frequency burst firing in supragranular cortical neurons," *Journal of Neuroscience*, vol. 20, no. 13, pp. 4829–4843, 2000.

[33] B. W. Connors and M. J. Gutnick, "Intrinsic firing patterns of diverse neocortical neurons," *Trends in neurosciences*, vol. 13, no. 3, pp. 99–104, 1990.

[34] F. Rieke, D. Warland, R. d. R. Van Steveninck, W. S. Bialek *et al.*, *Spikes: exploring the neural code*. MIT press Cambridge, 1999, vol. 7, no. 1.

[35] E. D. Adrian and Y. Zotterman, "The impulses produced by sensory nerve-endings: Part ii. the response of a single end-organ," *The Journal of physiology*, vol. 61, no. 2, pp. 151–171, 1926.

[36] S. J. Thorpe, "Spike arrival times: A highly efficient coding scheme for neural networks," *Parallel processing in neural systems*, pp. 91–94, 1990.

[37] R. W. Friedrich and G. Laurent, "Dynamic optimization of odor representations by slow temporal patterning of mitral cell activity," *Science*, vol. 291, no. 5505, pp. 889–894, 2001.

[38] M. Stopfer, V. Jayaraman, and G. Laurent, "Intensity versus identity coding in an olfactory system," *Neuron*, vol. 39, no. 6, pp. 991–1004, 2003.

[39] R. F. Galán, S. Sachse, C. G. Galizia, and A. V. Herz, "Odor-driven attractor dynamics in the antennal lobe allow for simple and rapid olfactory pattern classification," *Neural computation*, vol. 16, no. 5, pp. 999–1012, 2004.

[40] V. Cutsuridis, S. Cobb, and B. P. Graham, "Encoding and retrieval in a model of the hippocampal ca1 microcircuit," *Hippocampus*, vol. 20, no. 3, pp. 423–446, 2010.

[41] A. Lansner, "Associative memory models: from the cell-assembly theory to biophysically detailed cortex simulations," *Trends in neurosciences*, vol. 32, no. 3, pp. 178–186, 2009.

[42] G. Palm, "Neural associative memories and sparse coding," *Neural Networks*, vol. 37, pp. 165–171, 2013.

[43] P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer, "Real-time classification and sensor fusion with a spiking deep belief network," *Frontiers in neuroscience*, vol. 7, p. 178, 2013.

[44] B. Rueckauer, I.-A. Lungu, Y. Hu, and M. Pfeiffer, "Theory and tools for the conversion of analog to spiking convolutional neural networks," *arXiv preprint arXiv:1612.04052*, 2016.

[45] S. M. Bohte, J. N. Kok, and H. La Poutre, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1-4, pp. 17–37, 2002.

[46] J. Xin and M. J. Embrechts, "Supervised learning with spiking neural networks," in *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, vol. 3. IEEE, 2001, pp. 1772–1777.

[47] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers in neuroscience*, vol. 10, p. 508, 2016.

[48] F. Zenke and S. Ganguli, "Superspike: Supervised learning in multilayer spiking neural networks," *Neural computation*, vol. 30, no. 6, pp. 1514–1541, 2018.

[49] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," *Advances in Neural Information Processing Systems*, pp. 1412–1421, 2018.

[50] P. Gopalakrishnan, D. Kanevsky, A. Nadas, and D. Nahamoo, "A generalization of the baum algorithm to rational objective functions," in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on.* IEEE, 1989, pp. 631–634.

[51] O. Chatterjee and S. Chakrabartty, "Decentralized global optimization based on a growth transform dynamical system model," *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

Appendix A

Remapping synaptic interactions in a standard spiking network: The synaptic integration of simple adaptations of spiking neural networks [15], without leakage, can be represented by

$$v_{i,n} = v_{i,n-1} + \sum_{i=1}^{M} W_{ij}\theta(v_{j,n}) + y_{i,n}, \tag{A.1}$$

where $\theta(v_{j,n})$ is an input spike from the j-th neuron at the n-th time-step and y_i is the depolarization due to an external current stimulus. When the membrane potential exceeds a threshold V_{th}, a spike occurs and the voltage is reset below threshold. Ignoring leakage when the potential is building up, we can rewrite (A.1) for successive time-steps $n = 1, 2, ..., N$ as follows

$$\begin{aligned} v_{i,1} &= v_{i,0} + \sum_{i=1}^{M} W_{ij}\theta(v_{j,1}) + y_{i,1} \\ v_{i,2} &= v_{i,1} + \sum_{i=1}^{M} W_{ij}\theta(v_{j,2}) + y_{i,2} \\ &. \\ &. \\ v_{i,N} &= v_{i,N-1} + \sum_{i=1}^{M} W_{ij}\theta(v_{j,N}) + y_{i,N} \end{aligned}$$

Summing over the time-steps and dividing by the total number of time-steps, we get

$$\frac{1}{N}v_{i,N} = \frac{1}{N}v_{i,0} + \sum_{i=1}^{M} W_{ij}\mathcal{E}_n[\theta(v_{j,n})] + \mathcal{E}_n[y_{i,n}]. \tag{A.2}$$

As $N \to \infty$, the first two terms approach zero, so that we can write (A.2) as

$$\sum_{j=1}^{M} W_{ij}\mathcal{E}_n[\theta(v_{j,n})] + \mathcal{E}_n[y_{i,n}] \to 0. \tag{A.3}$$

With a slight re-arrangement of terms, (A.3) can be written as

$$\mathcal{E}_n[\theta(v_{i,n})] = \sum_{j=1}^{M} W_{ij}\mathcal{E}_n[\theta(v_{j,n})] + \mathcal{E}_n[y_{i,n}], \tag{A.4}$$

where W_{ii} on the R.H.S. has been adjusted for the additional term on the L.H.S. The time-expectation of the spiking activity $\theta(.)$ is nothing but a measure of the mean firing rate, which we denote by r. Replacing $\mathcal{E}_n[\theta(v_{i,n})]$ by r_i we get,

$$r_i = \sum_{j=1}^{M} W_{ij}r_j + \bar{y}_i. \tag{A.5}$$

(A.5) is a simple linear rate-based network where the firing rate for the i-th neuron changes over time based on the firing rates of pre-synaptic neurons and the average external input. Variations of (A.5) may take into consideration the reset after a neuron has fired, allowing us to rewrite (A.5) as

$$r_i = \Phi(\sum_{j=1}^{M} W_{ij}r_j + \bar{y}_i), \tag{A.6}$$

where $\Phi(.)$ is a non-linearity that captures the saturation in firing rate. Additionally, introducing a leakage term in (A.1) leads to

$$r_i = \Phi(\sum_{j=1}^{M} W_{ij}s_j + \bar{y}_i), \tag{A.7}$$

where s is a synaptic filter that captures the rise and decay of post-synaptic currents due to pre-synaptic spikes.

Appendix B

Proof of Proposition I: For the sake of accessibility, we first present the proof of convergence of the dynamical system model using scalar functions and scalar variables before extending it to the domain of vectors.

Proof. We can decompose the scalar variable v_n as $v_n = v_n^+ - v_n^-$, where $v_n^+, v_n^- \geq 0$. The following additional constraint

$$v_n^+ + v_n^- = 1 \tag{B.1}$$

imposed on the variables v_n^+ and v_n^- would then always ensure that (10) is satisfied. Then we can write

$$\underset{|v|\leq 1}{\arg\min}\ \mathcal{H}(v) \equiv \underset{v^+ + v^- = 1;\ v^+, v^- \geq 0}{\arg\min} \mathcal{H}(v^+ - v^-). \tag{B.2}$$

We can solve the equivalent optimization problem under the linear constraint specified by (B.1) and the non-negativity constraints, using an iterative multiplicative update called Growth Transforms - a fixed point algorithm for optimizing any Lipschitz continuous objective function under similar equality constraints [50, 51].

The dicrete-time growth transform-based update equations for the new optimization variables for minimizing $\mathcal{H}$ are given by

$$v_{n+1}^+ = \frac{v_n^+}{\mu_n}(-\frac{\partial \mathcal{H}}{\partial v_n^+} + \lambda) \tag{B.3}$$

$$v_{n+1}^- = \frac{v_n^-}{\mu_n}(-\frac{\partial \mathcal{H}}{\partial v_n^-} + \lambda), \tag{B.4}$$

where

$$\mu_n = v_n^+(-\frac{\partial \mathcal{H}}{\partial v_n^+} + \lambda) + v_n^-(-\frac{\partial \mathcal{H}}{\partial v_n^-} + \lambda) \tag{B.5}$$

is a normalization factor that ensures $v_{n+1}^+ + v_{n+1}^- = 1$. From (B.3) and (B.4), using $v_n = v_n^+ - v_n^-$ and (B.1), we can easily show

$$v_{n+1} = \frac{-\frac{\partial \mathcal{H}}{\partial v_n} + \lambda v_n}{-v_n \frac{\partial \mathcal{H}}{\partial v_n} + \lambda}, \tag{B.6}$$

where we have used the relation $\frac{\partial \mathcal{H}}{\partial v} = \frac{\partial \mathcal{H}}{\partial v^+} = -\frac{\partial \mathcal{H}}{\partial v^-}$. This dynamical system model, derived from the Growth Transform updates outlined in (B.3) and (B.4), ensures that 11 holds, with equality being satisfied iff v_n is a critical point of $\mathcal{H}$.

Rearranging the terms in (B.6), we get

$$(1 - v_n v_{n+1})\frac{\partial \mathcal{H}}{\partial v_n} + \lambda(v_{n+1} - v_n) = 0. \tag{B.7}$$

Taking expectation over time-indices on both sides, as $n \to \infty$, we have

$$\mathcal{E}_n\Big[(1 - v_n v_{n+1})\frac{\partial \mathcal{H}}{\partial v_n}\Big] \to 0. \tag{B.8}$$

This guarantees that if v_n is close to 0, the gradient $\frac{\partial \mathcal{H}}{\partial v_n}$ asymptotically goes to zero.

Extending this to a vector dynamical system where $\mathcal{H} : \mathcal{R}^M \to \mathcal{R}$ is a function of $\boldsymbol{v}_n =$

$\{v_{1,n}, v_{2,n}, .., v_{M,n}\}$, we can write

$$v_{i,n+1} = \frac{-\frac{\partial \mathcal{H}}{\partial v_{i,n}} + \lambda v_{i,n}}{-v_{i,n}\frac{\partial \mathcal{H}}{\partial v_{i,n}} + \lambda}, \quad i = 1, ..., M. \tag{B.9}$$

Here, λ is admissible iff $\lambda > |\frac{\partial \mathcal{H}}{\partial v_{i,n}}| \ \forall i, n$ [50].

Appendix C

Continuous-time Growth Transform dynamical system: In order to derive the complete dynamical system model for the Growth Transform neuron, we apply an useful property of Growth Transforms. The Growth Transform mapping $\sigma(.)$ homotopically minimizes the value of the cost function $\mathcal{H}$ ([23]) as shown below

$$\mathcal{H}\Big((1 - f_{i,n})v_{i,n}^{\gamma} + f_n \sigma(v_{i,n}^{\gamma})\Big) \leq \mathcal{H}\Big(v_{i,n}^{\gamma}\Big), \quad \gamma = +, -; \tag{C.1}$$

where $0 \leq f_{i,n} \leq 1$. This leads to the updated discrete-time equations for the new optimization variables for minimizing $\mathcal{H}$

$$v_{i,n+1}^{+} = (1 - f_{i,n})v_{i,n}^{+} + f_{i,n}\frac{v_{i,n}^{+}}{\mu_n}(-\frac{\partial \mathcal{H}}{\partial v_{i,n}^{+}} + \lambda) \tag{C.2}$$

$$v_{i,n+1}^{-} = (1 - f_{i,n})v_{i,n}^{-} + f_{i,n}\frac{v_{i,n}^{-}}{\mu_n}(-\frac{\partial \mathcal{H}}{\partial v_{i,n}^{-}} + \lambda). \tag{C.3}$$

From (C.2) and (C.3), following similar steps as before, we can show that

$$v_{i,n+1} = (1 - f_{i,n})v_{i,n} + f_{i,n}\frac{-\frac{\partial \mathcal{H}}{\partial v_{i,n}} + \lambda v_{i,n}}{-v_{i,n}\frac{\partial \mathcal{H}}{\partial v_{i,n}} + \lambda}. \tag{C.4}$$

Rearranging the terms in (C.4), we can write

$$\begin{aligned} \frac{1}{f_{i,n}}\left[v_{i,n+1} - v_{i,n}\right] + v_{i,n} &= \frac{-\frac{\partial \mathcal{H}}{\partial v_{i,n}} + \lambda v_{i,n}}{-v_{i,n}\frac{\partial \mathcal{H}}{\partial v_{i,n}} + \lambda}, \quad \text{or} \\ \frac{1}{f_{i,n}}\Big(\frac{\Delta v_{i,n}}{(n+1) - n}\Big) + v_{i,n} &= \frac{-\frac{\partial \mathcal{H}}{\partial v_{i,n}} + \lambda v_{i,n}}{-v_{i,n}\frac{\partial \mathcal{H}}{\partial v_{i,n}} + \lambda}. \end{aligned} \tag{C.5}$$

When the difference between successive time-steps is infinitesimally small, this reduces to the following continuous-time dynamical system model ([51])

$$\frac{1}{f_i(t)}\frac{dv_i(t)}{dt} + v_i(t) = \frac{-\frac{\partial \mathcal{H}}{\partial v_i(t)} + \lambda v_i(t)}{-v_i(t)\frac{\partial \mathcal{H}}{\partial v_i(t)} + \lambda}. \tag{C.6}$$