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

In neuromorphic engineering, neural populations are generally modeled in a bottom-up manner, where individual neuron models are connected through synapses to form large-scale spiking networks. Alternatively, a top-down approach treats the process of spike generation and neural representation of excitation in the context of minimizing some measure of network energy. However, these approaches usually define the energy functional in terms of some statistical measure of spiking activity (ex. firing rates), which does not allow independent control and optimization of neurodynamical parameters. In this paper, we introduce a new spiking neuron and population model where the dynamical and spiking responses of neurons can be derived directly from a network objective or energy functional of continuous-valued neural variables like the membrane potential. The key advantage of the model is that it 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 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 upon convergence to the optimal solution. 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

Spiking neural networks that emulate neural ensembles have 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 of excitation 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]. 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 [8], 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. Each neuron in the network implements an asynchronous mapping based on polynomial Growth Transforms, which is a fixed-point algorithm for optimizing polynomial functions under linear and/or bound constraints [9, 10]. It was shown in [8] 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 an optimization problem. We show that this network objective function or energy functional can be interpreted as the total extrinsic power required by the remapped network to operate, and hence a metric to be minimized.
- We then introduce a dynamical system model based on Growth Transforms that evolves the network towards this steady-state attractor under the specified constraints. 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 show how gradient discontinuity in the regularization function of the network energy functional can be used to incorporate the shape of the action potential while maintaining the local convexity of the energy functional and hence the location of the steady-state attractor.
- 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. We also illustrate how decoupling transient spiking dynamics from the network solution and spike-shapes could be

beneficial by using the model to design a spiking associative memory network that 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

In this section, we present 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. 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 $\boldsymbol{x}$ where $n = 1, 2, ..$ denotes a discrete time step
$x_i(t)$	i-th element of real-valued vector $\boldsymbol{x}$ at time t
$x_i[n]$	a sequence of scalar variables $x_{i,p}$ where $p = n, n-1, ..$
$\mathbb{E}_N\Big(x_i[n]\Big)$	Empirical expectation of a sequence $x_i[n]$ estimated over a window of size N, i.e., $\frac{1}{N}\sum\limits_{p=n-N+1}^{n} x_{i,p}$
$\bar{x}_i[n]$	Empirical expectation estimated over an asymptotically infinite window, i.e., $\lim\limits_{N\to\infty} \mathbb{E}_N\Big(x_i[n]\Big)$
$\bar{\boldsymbol{x}}[n]$	Real-valued vector with $\bar{x}_i[n]$ as its i-th element
$\mathbb{E}_T[x_i(t)]$	Empirical expectation of $x_i(t)$ over a time-interval $[t-T, t]$, i.e., $\frac{1}{T}\int_{t-T}^{t} x_i(t')dt'$
$\mathbb{R}^M$	Vector space spanned by M-dimensional real vectors
$\lvert x\rvert$	Absolute value of a scalar
$\lvert\lvert\boldsymbol{x}\rvert\rvert_p$	l_p-norm of an M-dimensional vector, defined as $(\sum\limits_{i=1}^{M} \lvert x_i\rvert^p)^{1/p}$
$\boldsymbol{x}^T$	Transpose of the vector $\boldsymbol{x}$
$\boldsymbol{x}.\boldsymbol{y}$	Inner product between the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$
$\frac{\partial\mathcal{H}}{\partial\boldsymbol{x}}$	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 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]. We consider a similar model where $v_i \in \mathbb{R}$ represents an inter-cellular membrane potential corresponding to

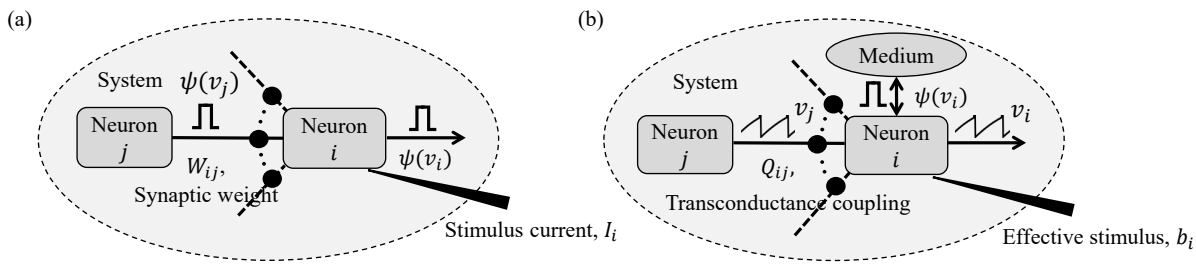

Figure 1: (a) Simple but general model of a spiking neural network; (b) Compartmental network model obtained after remapping.

neuron i in a network of M neurons. The i-th neuron receives spikes from the j-th neuron that are modulated by a synapse whose strength or weight is denoted by $W_{ij} \in \mathbb{R}$. Assuming that the synaptic weights are constant, the following discrete-time temporal equation governs the dynamics when the membrane potential increases [16, 17]

$$v_{i,n+1} = \gamma v_{i,n} + \sum_{j=1}^{M} W_{ij}\Psi(v_{j,n}) + y_{i,n}, \;\; \forall i = 1, ..., M, \tag{1}$$

where $v_{i,n} \equiv v_i(n\Delta t)$ and $v_{i,n+1} \equiv v_i\big((n+1)\Delta t\big)$, Δt being the time increment between two time-steps. $y_{i,n}$ represents the depolarization due to an external stimulus that can be viewed as $y_{i,n} = R_{mi}I_{i,n}$, where $I_{i,n} \in \mathbb{R}$ is the current stimulus at the n-th time-step and $R_{mi} \in \mathbb{R}$ is the membrane resistance of the i-th neuron. Here, $0 \leq \gamma \leq 1$ denotes the leakage factor and $\Psi(.)$ denotes a simple spiking function that is positive only when the voltage $v_{j,n}$ exceeds a threshold and 0 otherwise. Note that in (1), the filter $\Psi(.)$ implicitly depends on the pre-synaptic spike-times through the pre-synaptic membrane voltage $v_{j,n}$. Such a spiking neural network model is shown in Figure 1(a).

We further enforce that the membrane potentials are bounded by v_c as

$$|v_{i,n}| \leq v_c, \;\; \forall i = 1, ..., M, \;\; \forall n. \tag{2}$$

Note that in biological neural networks, the membrane potentials are also bounded [18].

If $\Psi(.)$ was a smooth function of the membrane potential, $v_{i,n}$ would track the net input at every instant. For a non-smooth $\Psi(.)$, however, we make the additional assumption that the temporal expectation of $v_{i,n}$ encodes the net input over a sufficiently large time-window. Considering $\bar{y}_i[n]$ to be the empirical expectation of the external input estimated at the n-th time-window, and under the bound constraints outlined in (2), we can get the following relation (justification in Appendix A)

$$(1-\gamma)\bar{v}_i[n] = \sum_{j=1}^{M} W_{ij}\bar{\Psi}_j[n] + \bar{y}_i[n], \tag{3}$$

where $\bar{\Psi}_j[n] = \lim_{N\to\infty} \sum_{p=n-N+1}^{n} \Psi(v_{j,p})$. To reduce notational clutter, we will re-write (3) in a matrix form as

$$(1-\gamma)\bar{\boldsymbol{v}}[n] = \mathbf{W}\bar{\boldsymbol{\Psi}}[n] + \bar{\boldsymbol{y}}[n], \tag{4}$$

where $\bar{\boldsymbol{v}}[n] \in \mathbb{R}^M$ is the vector of mean membrane potentials for a network of M neurons, $\mathbf{W} \in \mathbb{R}^M \times \mathbb{R}^M$ is the synaptic weight matrix for the network, $\bar{\boldsymbol{y}}[n] \in \mathbb{R}^M$ is the vector of mean external inputs for the n-th time-window and $\bar{\boldsymbol{\Psi}}[n] = \big[\bar{\Psi}_1[n], \bar{\Psi}_2[n], ..., \bar{\Psi}_M[n]\big]^T$ is the vector of mean spike currents. As $\Psi(.)$ is a non-linear function of the membrane potential, 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

$$\bar{\boldsymbol{\Psi}}[n] = (1-\gamma)\mathbf{W}^{-1}\bar{\boldsymbol{v}}[n] - \mathbf{W}^{-1}\bar{\boldsymbol{y}}[n], \text{ or} \tag{5}$$

$$\bar{\boldsymbol{\Psi}}[n] = -\mathbf{Q}\bar{\boldsymbol{v}}[n] + \bar{\boldsymbol{b}}[n], \tag{6}$$

where $\mathbf{Q} = -(1-\gamma)\mathbf{W}^{-1}$, and $\bar{\boldsymbol{b}}[n] = -\mathbf{W}^{-1}\bar{\boldsymbol{y}}[n]$ 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

$$\bar{\Psi}_i[n] = -\sum_{j=1}^{M} Q_{ij}\bar{v}_j[n] + \bar{b}_i[n], \tag{7}$$

subject to the bound constraint $|v_{i,n}| \leq v_c \ \forall i, n$. In the subsequent sections, we show that (7) can be viewed as the first-order condition of the following network objective function or energy functional

$$\min_{|v_i|\leq v_c \ \forall i} \mathcal{H}\big(\{v_i\}\big) = \min_{|v_i|\leq v_c \ \forall i} \ \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} Q_{ij}v_jv_i - \sum_{i=1}^{M} b_iv_i + \sum_{i=1}^{M}\int_{-\infty}^{v_i} \Psi(v)dv. \tag{8}$$

The network energy functional $\mathcal{H}(.)$ in (8) also admits a physical interpretation, as shown in Figure 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]. The neuron i also receives an electrical current stimulus b_i and exchanges a 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.

2.2 Neuron model using the Growth Transform dynamical system

In order to solve the energy minimization problem given in (8) under the constraints given in (2), we first propose a dynamical system based on polynomial Growth Transforms. We also show how the dynamical system evolves over time to satisfy (7) as a first-order condition.

Growth Transforms are multiplicative updates derived from the well-known Baum-Eagon inequality [9, 20] that optimize a Lipschitz continuous cost function under linear and/or bound constraints on the optimization variables. Each neuron in the network implements a continuous mapping based on Growth Transforms, ensuring that the network evolves over time to reach an optimal solution of the energy functional within the constraint manifold. The summary of the proposed model is presented in Table 1 and the detailed derivation is provided in Appendix B.

2.2.1 Growth Transform spiking neuron model

Considering the n-th iteration of the update equation in (9) as the n-th time-step for the neuron i, we can rewrite (9) in terms of the objective function for the neuron model presented in (8), as given below

$$v_{i,n+1} \leftarrow v_c \frac{-\frac{\partial \mathcal{H}}{\partial v_{i,n}} v_c + \lambda v_{i,n}}{-\frac{\partial \mathcal{H}}{\partial v_{i,n}} v_{i,n} + \lambda v_c}, \ \ i = 1, ..., M, \tag{13}$$

where

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

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

Proposition I. Let $\mathcal{H}\big(\{v_i\}\big) : \mathbb{R}^M \to \mathbb{R}$ be a function of $v_i,\ \ i = 1, ..., M$ with bounded partial derivatives, and let $\lambda > |\frac{\partial \mathcal{H}}{\partial v_{i,n}}|\ \ \forall i, n$, be a parameter. Then for $|v_{i,0}| \leq v_c\ \ \forall i$, the discrete-time dynamical system

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

satisfies the following criteria for all time-indices n:

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

$$(b)\ \ \mathcal{H}\big(\{v_{i,n+1}\}\big) \leq \mathcal{H}\big(\{v_{i,n}\}\big)\ \text{ in domains where } \frac{\partial \mathcal{H}}{\partial v_{i,n}} \text{ is continuous; and} \tag{11}$$

$$(c)\ \lim_{N\to\infty} \Big(\mathbb{E}_N\Big(z_i[n]\Big)\Big) \to 0\ \ \forall i, n;\ \text{ where } z_{i,n} = (v_c^2 - v_{i,n} v_{i,n+1}) \frac{\partial \mathcal{H}}{\partial v_{i,n}}. \tag{12}$$

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

$$\lim_{N\to\infty} \Big(\mathbb{E}_N\Big(z_i[n]\Big)\Big) \to 0\ \ \forall i, n, \tag{15}$$

where $z_{i,n} = (v_c^2 - v_{i,n} v_{i,n+1}) \frac{\partial \mathcal{H}}{\partial v_{i,n}}$. We first show the dynamics resulting from (13) for a trivial barrier function $\Psi(.) = 0$. Since $\mathcal{H}(.)$ is a smooth function in this case, the neural variables $v_{i,n}$ converge to a local minimum, such that

$$\lim_{n\to\infty} v_{i,n} = v_i^*, \tag{16}$$

Therefore, (15) can be written as

$$(v_c^2 - v_i^{*2}) \frac{\partial \mathcal{H}}{\partial v_{i,n}}\Big|_{v_i^*} \to 0. \tag{17}$$

Thus as long as $|v_i^*| < v_c$, the gradient term goes to zero, ensuring that the dynamical system converges to the optimal solution within the domain defined by the bound constraints.

The dynamical system presented in (9) ensures that the steady-state neural responses $|v_i^*| \leq v_c\ \forall i$. In the absence of the barrier term, the membrane potentials can converge to any value between $-v_c$ and $+v_c$ based on the effective inputs to individual neurons. Figure 2(a) illustrates this for 2 different neurons where $\mathbf{Q}$ is an identity matrix. For the sake of simplicity, we have considered the membrane potentials to be normalized in all the experiments in this paper (i.e., $v_c = 1\ V$), and $0\ V$ as the threshold voltage. Here v_1^* is hyperpolarized due to a negative stimulus, and v_2^* is depolarized beyond the threshold. Figure 2(b) shows the corresponding energy contours, where the steady-state neural responses encode the optimal solution of the energy function. We next show how this framework can be extended to a spiking neuron model when the trans-membrane current in the compartmental model described in (8) is approximated by a discontinuous $\Psi(.)$. In general, the penalty function $R(v_i) = \int_{-\infty}^{v_i} \Psi(v) dv$ is chosen to be convex, where $R(v_i > 0\ V) > 0\ W$ and $R(v_i \leq 0\ V) = 0\ W$. Figure 2(c) shows one such form that has a gradient discontinuity at a threshold ($v_i = 0\ V$) at which the neuron generates an

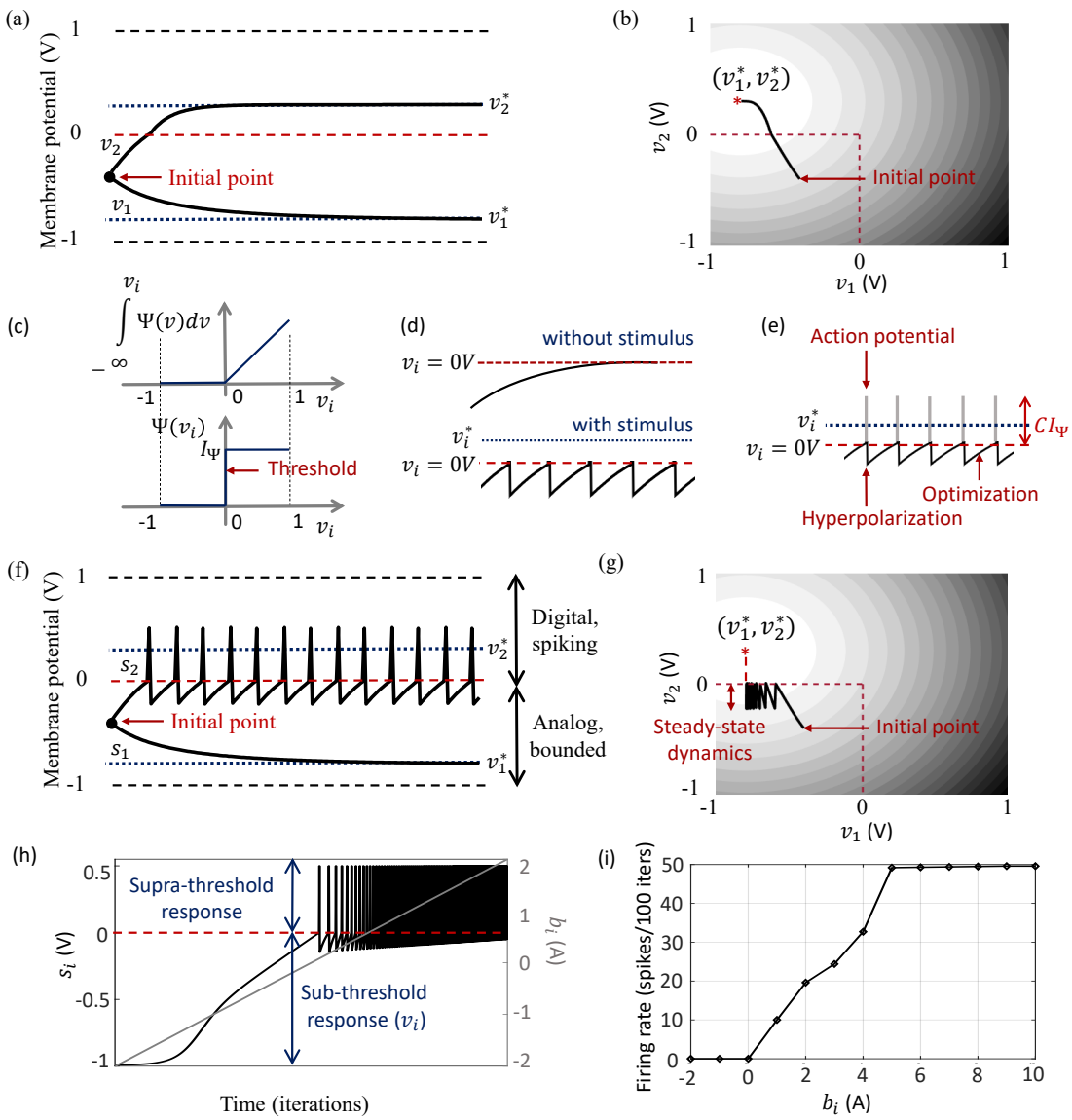

Figure 2: (a) Bounded dynamics in a 2-neuron network in absence of the barrier function; (b) Corresponding contour plot showing convergence of the membrane potentials in the presence of external stimulus; (c) The function $\int \Psi(.)dv$ and its derivative $\Psi(.)$ used in this paper for the spiking neuron model; (d) Time-evolution of the response v_i of a single neuron in the spiking model in the absence and presence of external stimulus; (e) The composite signal upon addition of spikes when v_i crosses the threshold; (f) Bounded and spiking dynamics in the same 2-neuron network in presence of barrier function; (g) Corresponding contour plot showing steady-state dynamics of the membrane potentials in the presence of external stimulus; (h) Plot of composite spike signal s_i of the spiking neuron model when the external current stimulus is increased; (i) Input-output characteristics for the spiking neuron model.

action potential. The corresponding $\Psi(.)$, also shown in Figure 2(c), is given by

$$\Psi(v_{i,n}) = \begin{cases} I_\Psi \ A & ; \quad v_{i,n} > 0 \ V \\ 0 \ A & ; \quad v_{i,n} \leq 0 \ V. \end{cases} \tag{18}$$

When there is no external stimulus b_i, the neuron response converges to $v_i^* = 0 \ V$ as in the non-spiking case, as illustrated in Figure 2(d) for a single neuron without any synaptic connections. 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 Figure 2(d). However, a penalty term $R(v_i)$ of the form as described above works as a barrier function, penalizing the energy functional whenever v_i exceeds the threshold, thereby forcing v_i to reset below the threshold. The stimulus and the barrier function therefore introduce opposing tendencies, making v_i oscillate back and forth around the discontinuity (which, in our case, coincides with the threshold) as long as the stimulus is present. Thus when $\Psi(.)$ is introduced, the potential $v_{i,n}$ switches when $\Psi(v_{i,n}) > 0 \ A$ or only when $v_{i,n} > 0 \ V$. However, the dynamics of $v_{i,n}$ remains unaffected for $v_{i,n} < 0 \ V$. During the brief period when $v_{i,n} > 0 \ V$, we assume that the neuron enters into a runaway state leading to a voltage spike. The composite spike signal $s_{i,n}$, shown in Figure 2(e), is then treated as a combination of the sub-threshold and supra-threshold responses and is given by

$$s_{i,n} = v_{i,n} + C\Psi(v_{i,n}), \tag{19}$$

where the trans-impedance parameter $C > 0 \ \Omega$ determines the magnitude of the spike. An expectation over the composite signal $s_{i,n}$ asymptotically encodes the new optimal solution v_i^* (dotted line), as shown in Figure 2(e) and later in Section 2.2.2. 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. Thus a refractory period is automatically incorporated in between two spikes.

In order to show the effect of $\Psi(.)$ on the nature of the solution, we plot the neural responses and contour plots for the 2-neuron network in Figures 2(f) and (g) for the same set of inputs as in Figures 2(a) and (b), when the barrier function is present. The penalty function produces a barrier at the thresholds, which are indicated by red dashed lines, transforming the time-evolution of s_2 into a digital, spiking mode, where the firing rate is determined by the extent to which the neuron is depolarized. It can be seen from the neural trajectories in Figure 2(g) and from (8) that $\Psi(.) > 0$ behaves as a Lagrange parameter corresponding to the spiking threshold constraint $v_{i,n} < 0$.

In Appendix C, we outline how, for non-pathological cases, it can be shown from (12) that for spiking neurons, or for neurons whose membrane potentials $v_{i,n} > -v_c \ \forall n$,

$$\lim_{N\to\infty} \left(\mathbb{E}_N \left(\frac{\partial \mathcal{H}}{\partial v_i}[n] \right) \right) = 0, \tag{20}$$

This 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 [21] to derive a dynamical system corresponding to $\Sigma\Delta$ modulation for tracking low-dimensional manifolds embedded in high-dimensional analog signal spaces. Combining (14) and (20), we have

$$\sum_{j=1}^{M} Q_{ij}\bar{v}_j[n] - \bar{b}_i[n] + \bar{\Psi}_i[n] = 0, \tag{21}$$

where $\bar{\Psi}_i[n] = \lim_{N\to\infty} \frac{1}{N} \sum_{p=n-N+1}^{n} \Psi(v_{i,p})$. Rearranging the terms in (21), we obtain (7).

The penalty function $R(v_i) = \int_{-\infty}^{v_i} \Psi(v)dv$ in the network energy functional in effect models the power dissipation due to spiking activity. For the form of $R(.)$ chosen in this paper, the power dissipation due to spiking is taken to be zero below the threshold, and increases linearly above threshold. A plot of the composite spike signal for a ramp input for the spiking neuron model is presented in Figure 2(h). As $v_{i,n}$ exceeds the threshold for a positive stimulus, the neuron enters a spiking regime and the firing rate increases with the input, whereas the sub-threshold response is similar to the non-spiking case. Figure 2(i) 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 $\boldsymbol{v} = \{v_i\} \in \mathbb{R}^M$, where the trans-conductance coupling matrix is denoted by $\mathbf{Q} = \{Q_{ij}\} \in \mathbb{R}^M \times \mathbb{R}^M$ and the external stimulus vector is denoted by $\boldsymbol{b} = \{b_i\} \in \mathbb{R}^M$, the time-evolution of the network under bound constraints on the state variables $|v_{i,n}| \leq v_c$ for all time-indices n, is governed by the following discrete-time updates:

$$v_{i,n+1} \leftarrow v_c \frac{-\frac{\partial \mathcal{H}}{\partial v_{i,n}} v_c + \lambda v_{i,n}}{-\frac{\partial \mathcal{H}}{\partial v_{i,n}} v_{i,n} + \lambda v_c}, \quad i = 1, ..., M, \tag{22}$$

where

$$\frac{\partial \mathcal{H}}{\partial v_{i,n}} = \sum_{j=1}^{M} Q_{ij} v_{j,n} - b_{i,n} + \Psi(v_{i,n});$$

$$\Psi(v_{i,n}) = \left\{ \begin{array}{lll} I_\Psi \ A & ; & v_{i,n} > 0 \ V \\ 0 \ A & ; & v_{i,n} \leq 0 \ V \end{array} \right\};$$

λ is a fixed current parameter such that $\lambda > |\frac{\partial \mathcal{H}}{\partial v_{i,n}}| \ \forall i, n$.

The composite spike response of the i-th neuron at time-step n is given by

$$s_{i,n} = v_{i,n} + C\Psi(v_{i,n}),$$

where the trans-impedance parameter $C > 0 \ \Omega$ determines the magnitude of each spike.

2.2.2 Encoding stimuli as a combination of sub-threshold and supra-threshold dynamics

As explained previously, the penalty term $R(v_i)$ of the form presented above works analogous to a barrier function, penalizing the energy functional whenever $v_{i,n}$ exceeds the threshold. This transforms the time-evolution of $v_{i,n}$ into a spiking mode above the threshold, while keeping the sub-threshold dynamics similar to the non-spiking case. The Growth Transform dynamical system ensures that the membrane potentials are bounded, thereby implementing a squashing (compressive) function on the neural responses so that the network responses are bounded. We now show how the proposed model encodes external stimulus as a combination of spiking and

bounded dynamics. In the steady-state, from (21) we can write

$$\bar{\Psi}_i[n] = \bar{b}_i[n] - \sum_{j=1}^{M} Q_{ij}\bar{v}_j[n]. \tag{23}$$

Thus the average spiking activity of the i-th neuron encodes the error between the average input and the weighted sum of membrane potentials. For a single, uncoupled neuron where

$$Q_{ij} = \begin{cases} Q_0\ \Omega^{-1} & ;\ \forall i = j \\ 0\ \Omega^{-1} & ;\ \forall i \neq j \end{cases}, \tag{24}$$

we have

$$\bar{\Psi}_i[n] + Q_0\bar{v}_i[n] = \bar{b}_i[n]. \tag{25}$$

Multiplying (25) on both sides by $C\ \Omega$, where we have chosen $C = \frac{1}{Q_0}$, we have

$$C\bar{\Psi}_i[n] + \bar{v}_i[n] = C\bar{b}_i[n] \tag{26}$$

$$\text{or,}\ \ s_i[n] = C\bar{b}_i[n], \tag{27}$$

where we have used the relation (19). (27) indicates that through a suitable choice of the trans-impedance parameter C, the sum of sub-threshold and supra-threshold responses encodes the external input to the neuron. This is also the rationale behind adding a spike to the sub-threshold response $v_{i,n}$, as illustrated in Figure 2(e), to yield the composite neural response. If $Q_0 = 0\ \Omega^{-1}$, we similarly have

$$\bar{\Psi}_i[n] = \bar{b}_i[n], \tag{28}$$

where the average spiking activity tracks the stimulus. Thus by defining the coupling matrix in various ways, we can obtain different encoding schemes for the network.

2.3 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 a geometric interpretation of neural dynamics. Similar to the network coding framework presented in [8], 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.

Like a Hessian, if we assume that the matrix $\mathbf{Q}$ is positive-definite about a local attractor, there exists a set of vectors $\boldsymbol{x}_i \in \mathbb{R}^D,\ \ i = 1, ..., M$ such that each of the elements Q_{ij} can be written as an inner product between two vectors as $Q_{ij} = \boldsymbol{x}_i.\boldsymbol{x}_j,\ \ 1 \leq i, j \leq M$. This is similar to kernel methods that compute similarity functions between pairs of vectors in the context of support vector machines [22]. This associates the i-th neuron in the network with a vector $\boldsymbol{x}_i$, mapping it onto an abstract metric space $\mathbb{R}^D$ and essentially providing an alternate geometric representation of the neural network. From (21), the spiking activity of the i-th neuron for the n-th time-window can then be represented as

$$\begin{aligned} \bar{\Psi}_i[n] &= -\sum_{j=1}^{M} Q_{ij}\bar{v}_j[n] + \bar{b}_i[n] \\ &= \sum_{j=1}^{M} -(\boldsymbol{x}_i.\boldsymbol{x}_j)\bar{v}_j[n] + \bar{b}_i[n] \\ &= \boldsymbol{w}_n.\boldsymbol{x}_i + \bar{b}_i[n], \end{aligned} \tag{29}$$

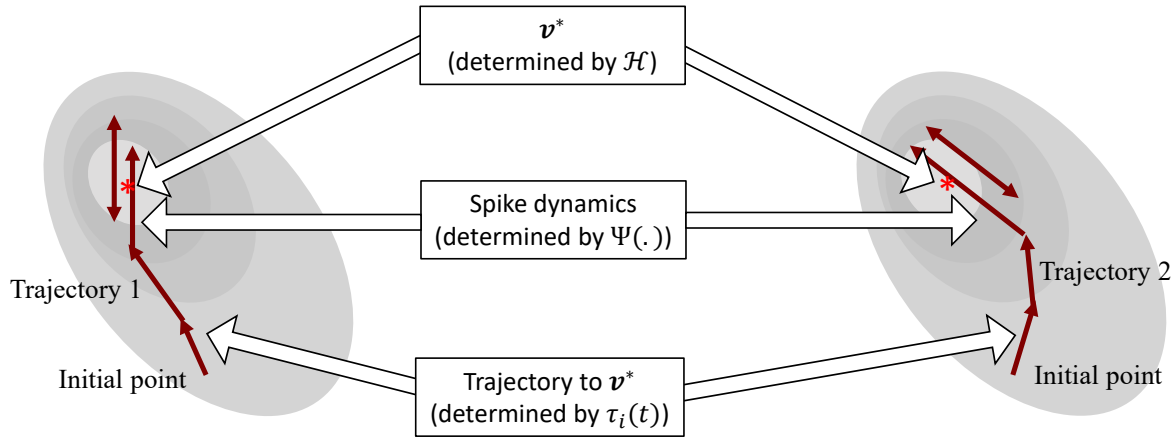

Figure 3: Decoupling of network solution, spike shape and response trajectory using the proposed model. Different modulation functions lead to different steady-state spiking dynamics under the same energy contour.

where

$$\boldsymbol{w}_n = -\sum_{j=1}^{M} \boldsymbol{x}_j \bar{v}_j[n]. \tag{30}$$

$\bar{\Psi}$ therefore represents the distance of the vector $\boldsymbol{x}_i$ from a network hyperplane in the D-dimensional vector space, which is parameterized by the weight vector $\boldsymbol{w}_n$ and offset $\bar{b}_i[n]$. When a stimulus $\bar{b}_i[n]$ 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 $\mathbf{Q}$. Hence, $\bar{\Psi}(.)$ 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.

2.4 Complete continuous-time 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 [23] provides an extensive review of different spiking neuron models and their ability to produce the different dynamics observed in biology. In this section, 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. In Appendix D, we derive the continuous-time version of the dynamical system using a special property of Growth Transforms. 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 a time-constant $\tau_i(t)$, also referred to as modulation function in this paper. Fortunately, the evolution of $\tau_i(t)$ 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 Figure 3. In the Results section, we show how different forms of $\tau_i(t)$ produce different sets of neuronal dynamics consistent with the dynamics that have been

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

For a network of M neurons with state variables $\boldsymbol{v} = \{v_i\} \in \mathbb{R}^M$, where the trans-conductance coupling matrix is denoted by $\mathbf{Q} = \{Q_{ij}\} \in \mathbb{R}^M \times \mathbb{R}^M$ and the external stimulus vector is denoted by $\boldsymbol{b} = \{b_i\} \in \mathbb{R}^M$, the time-evolution of the network under bound constraints on the state variables $v_i(t) \leq v_c \ \forall t$, is governed by the following continuous-time dynamical system:

$$\tau_i(t)\frac{dv_i(t)}{dt} + v_i(t) = v_c \frac{-\frac{\partial \mathcal{H}}{\partial v_i(t)} v_c + \lambda v_i(t)}{-\frac{\partial \mathcal{H}}{\partial v_i(t)} v_i(t) + \lambda v_c}, \tag{31}$$

where

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

$\Psi(v_i(t)) = \left\{ \begin{array}{lll} I_\Psi \ A & ; & v_i(t) > 0 \ V \\ 0 \ A & ; & v_i(t) \leq 0 \ V \end{array} \right\}$;

λ is a fixed current parameter such that $\lambda > |\frac{\partial \mathcal{H}}{\partial v_i(t)}| \ \forall i, t$;

$0 \leq \tau_i(t, v_i, \dot{v}_i, \mathcal{H}, \dot{\mathcal{H}}) < \infty$ 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 composite spike response of the i-th neuron at time t is given by

$$s_i(t) = v_i(t) + C\Psi(v_i(t)), \tag{32}$$

where the trans-impedance parameter $C > 0 \ \Omega$ determines the magnitude of each spike.

reported in neurobiology.

3 Results

The proposed approach enables us to decouple the three following aspects of the spiking neural network:

(a) Fixed points of the network energy functional, which depend on the network configuration and external inputs;

(b) Nature and shape of neural responses, without affecting the network minimum; and

(c) Spiking statistics and transient neural dynamics at the cellular level, without affecting the network minimum or spike shapes.

This makes it possible to independently control and optimize each of these neuro-dynamical properties without affecting the others. The first two aspects arise directly from an appropriate selection of the energy functional and were demonstrated in Section 2.2.1. In this section, we show how the modulation function, in essence, loosely models cell excitability, and can be varied to tune transient firing statistics based on local and/or global variables. This allows us to encode the same optimal solution using widely different firing patterns across the network, and have unique potential benefits for neuromorphic applications. Codes for the representative examples given in this section are available at [11].

3.1 Single-neuron dynamics

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

$$Q_{ij} = \begin{cases} Q_0\ \Omega^{-1} & , \quad \forall i = j \\ 0\ \Omega^{-1} & , \quad \forall i \neq j \end{cases} . \tag{33}$$

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, 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 parameters 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 [24, 25, 26]. The proposed model shows tonic spiking without adaptation when the modulation function $\tau_i(t) = \tau$, where $\tau > 0$ s. A simulation of tonic spiking response using the neuron model is given in Figure 4(a).

3.1.2 Bursting response

Bursting neurons fire discrete groups of spikes interspersed with periods of silence in response to a constant stimulus [24, 25, 27, 28]. Bursting arises from an interplay of fast ionic currents

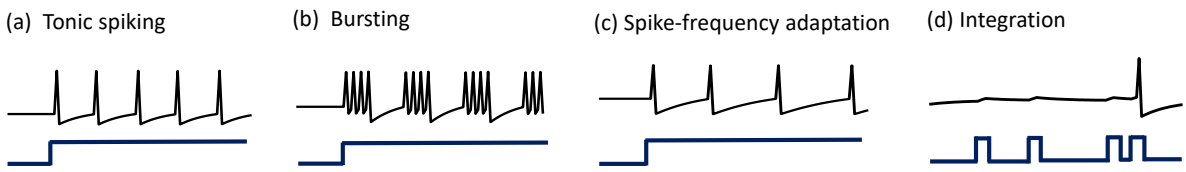

Figure 4: (a)-(d) Simulations demonstrating different single-neuron responses obtained using the GT neuron model.

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 $\tau_i(t)$ at a slower rate compared to the generation of action potentials, in the following way

$$\tau_i(t) = \begin{cases} \tau_1 \ s & , \quad c_i(t) < B \\ \tau_2 \ s & , \quad c_i(t) \geq B \end{cases} \tag{34}$$

where $\tau_1 > \tau_2 > 0$ s, B is a parameter and the count variable $c_i(t)$ is updated according to

$$c_i(t) = \begin{cases} \lim\limits_{\Delta t \to 0} c_i(t - \Delta t) + \mathcal{I}[v_i(t) > 0]) & , \quad \lim\limits_{\Delta t \to 0} c_i(t - \Delta t) < B \\ 0 & , \quad \lim\limits_{\Delta t \to 0} c_i(t - \Delta t) \geq B \end{cases} , \tag{35}$$

$\mathcal{I}[.]$ being an indicator function. Simulation of a bursting neuron in response to a step input is given in Figure 4(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 [29]. 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

$$\tau_i(t) = \tau - 2\phi\big(h(t) * \Psi(v_i(t))\big) \tag{36}$$

where $h(t) * \Psi(v_i)(t)$ is a convolution operation between a continuous-time first-order smoothing filter $h(t)$ and the spiking function $\Psi(v_i(t))$, and

$$\phi(x) = \tau\Big(\frac{1}{1 + \exp(x)}\Big) \tag{37}$$

is a compressive function that ensures $0 \leq \tau_i(t) \leq \tau$ s. The parameter τ determines the steady-state firing rate for a particular stimulus. A tonic-spiking response with spike-frequency adaptation is shown in Figure 4(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 Figure 4(d), preferentially spiking to high-frequency or closely-spaced input pulses that are more likely to make v_i cross the threshold.

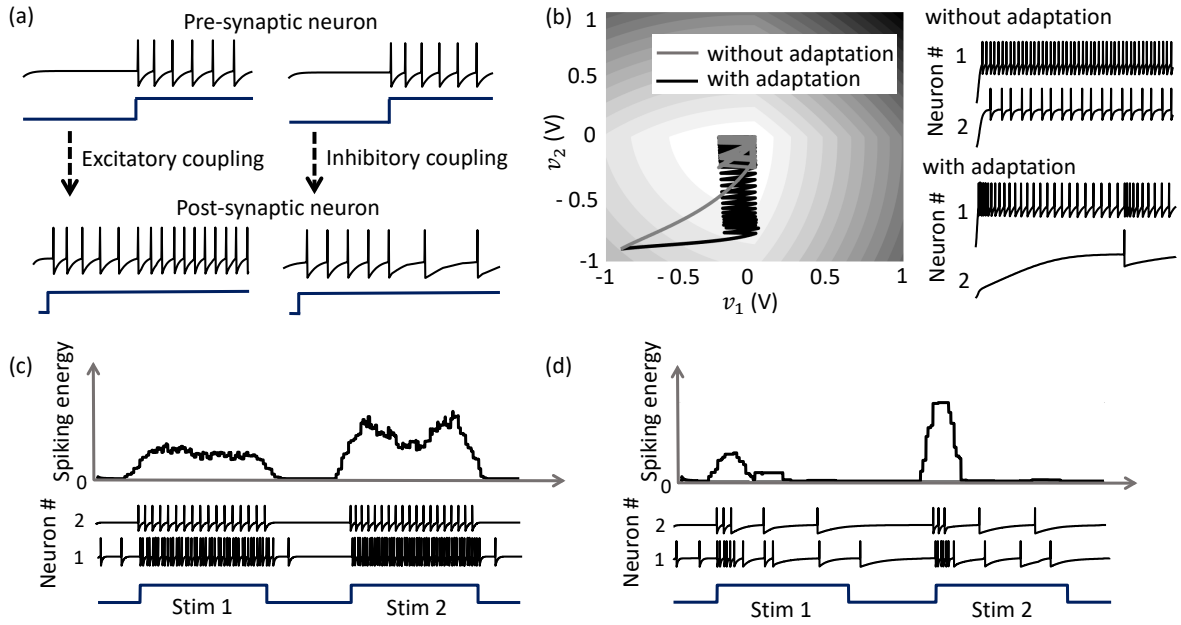

Figure 5: (a) Results from a 2-neuron network with excitatory and inhibitory couplings; (b) Energy optimization process under different conditions lead to different limit cycles within the same energy landscape. (c)-(d) Mean spiking energy $\int \Psi(.)dv$ and firing patterns in response to a step input without and with global adaptation respectively.

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 a unique solution of (8). Although elements of the coupling matrix $\mathbf{Q}$ already capture the interactions among neurons in a coupled network, we can further define the modulation function as follows to make the proposed model behave as a standard spiking network

$$\tau_i(t) = \phi\left(h(t) * \sum_{j=1}^{M} Q_{ij}\Psi(v_j(t))\right) \tag{38}$$

with the compressive-function $\phi(.)$ given by (37). (38) 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 Figure 5(a). Note that irrespective of whether such a pre-synaptic adaptation is implemented or not, the neurons under the same energy landscape would converge to the same sub-domain, albeit with different response trajectories and steady-state limit-cycles. This is illustrated in Figure 5(b) which plots the energy contours for a two-neuron network corresponding to a $\mathbf{Q}$ matrix with excitatory and inhibitory connections and a fixed stimulus vector $\boldsymbol{b}$. Figure 5(b) also shows 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 the only coupling between the two neurons is through the coupling matrix $\mathbf{Q}$, but there is no pre-synaptic spike-time dependent adaptation). 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 input spikes received by each neuron, neurons in the coupled network can be made to adapt according to the global dynamics by changing the modulation function as follows

$$\tau_i(t) = \phi\Big(h(t) * \big(\sum_{j=1}^{M} Q_{ij}\Psi(v_j(t)) - \mathcal{F}(\mathcal{H}, \dot{\mathcal{H}})\big)\Big) \tag{39}$$

with the compressive-function $\phi(.)$ given by (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 $\tau_i(.)$ adapts to reduce the spiking rate of the neuron without affecting the steady-state solution. Figures 5(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

$$\mathcal{F}(\mathcal{H}, \dot{\mathcal{H}}) = \begin{cases} \mathcal{F}_0 & , \quad \mathbb{E}_T(\dot{\mathcal{H}}) \approx 0 \\ 0 & , \quad \text{otherwise.} \end{cases} \tag{40}$$

where $\mathcal{F}_0 > 0$ is a tunable parameter. 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 the geometric interpretation outlined in Section 2.3, 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 $\mathbf{Q}$ 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 co-ordinate space, while placing neurons with weaker couplings far away from each other. A network consisting of 20 neurons is shown in Figure 6(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. Figure 6(b) shows the spike raster for the entire network. We see that the responsiveness of the neurons to a particular stimulus increases 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, 30]. 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 [31]. A temporal code like the time-to-first-spike posits that a stronger stimulus brings a neuron to the spiking threshold faster, generating a spike, and hence relative spike arrival times contain critical information about the stimulus [32].

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

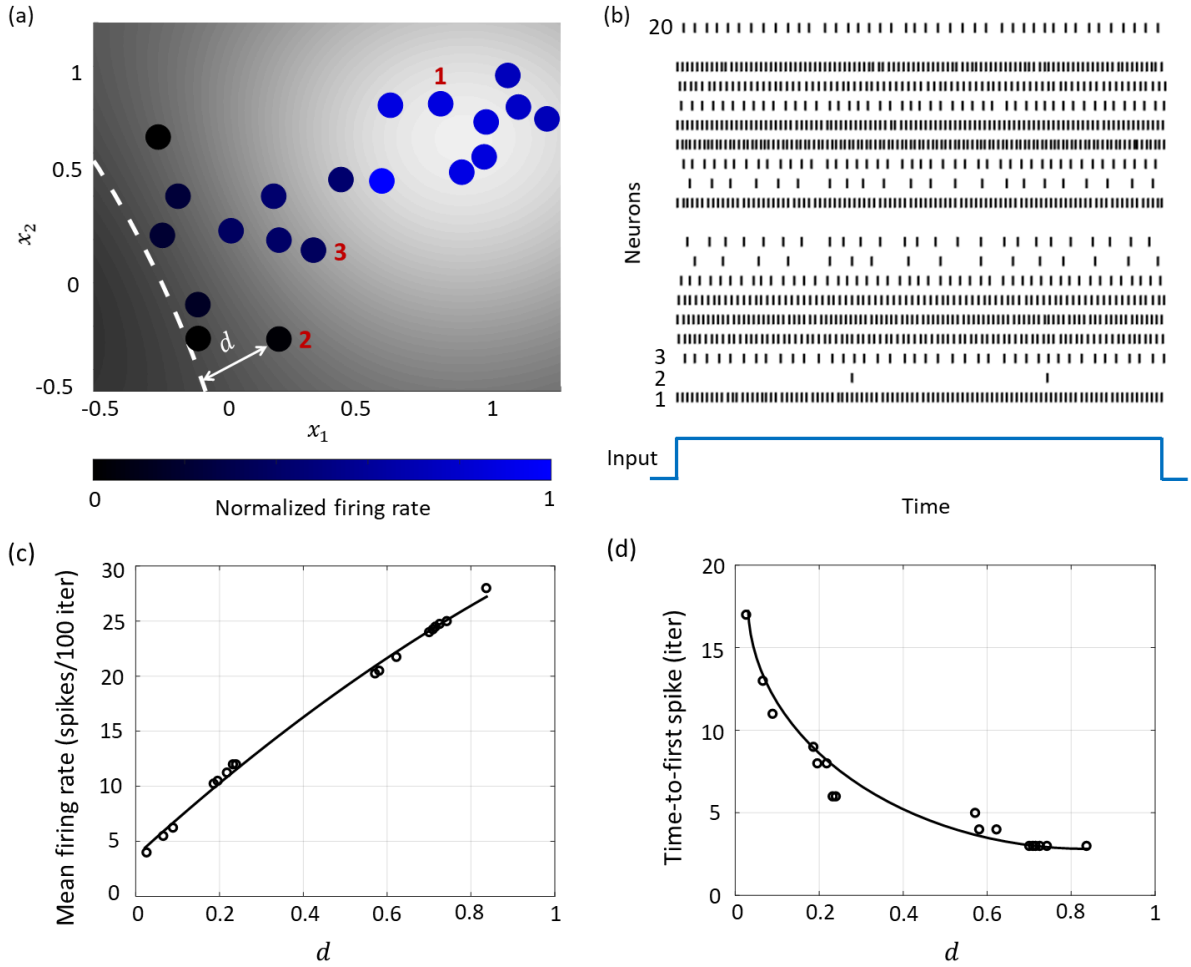

Figure 6: (a) Contour plot of spiking activity corresponding to a particular stimulus vector. Neurons are colored according to their mean firing rate (normalized w.r.t. the maximum firing rate) during the stimulus period. The white dashed line is the contour corresponding to $\bar{\Psi} = 0$. (b) Spike 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 d for each neuron in the network.

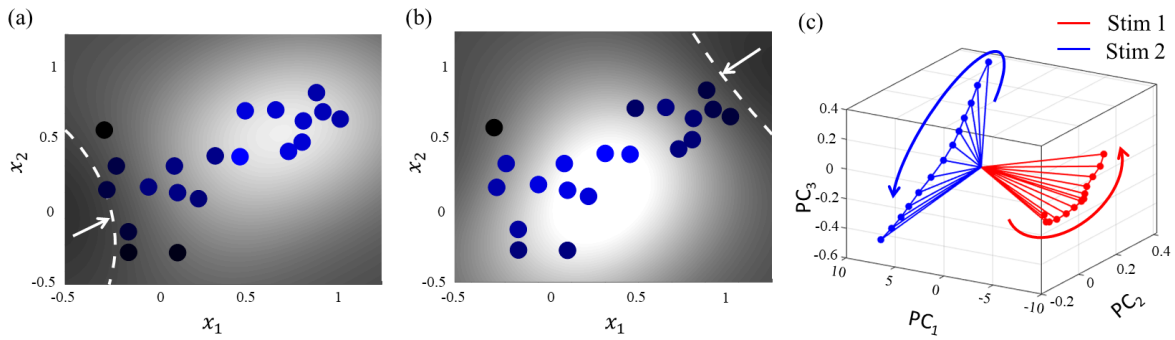

Figure 7: (a) and (b) Perturbation of the stimulus vector in different directions for the same network produces two different contours. (c) Corresponding population activities trace different trajectories in the neural subspace.

closely related to its proximity to the hyperplane. The neurons which exhibit more spiking are located at a greater distance from the hyperplane. We see from Figures 6(c) and (d) that as this value increases, the average firing rate of a neuron (number of spikes in a fixed number of time-steps or 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 in neurobiological literature by a 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 [33, 34]. 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 [33, 34, 35]. 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 Figures 7(a) and (b). A similar dimensionality reduction to three principal components in Figure 7(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.

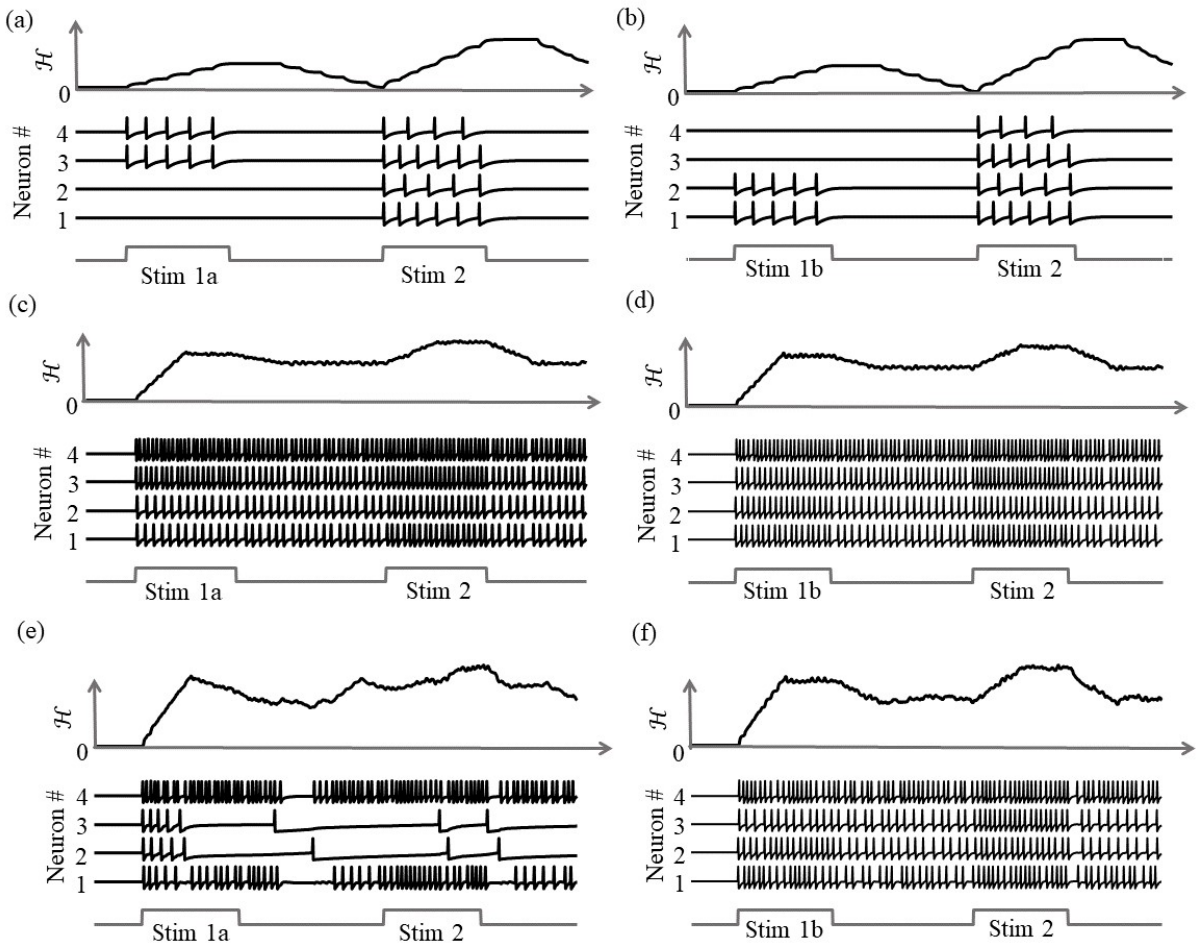

Figure 8: Stimulus response for a 4-neuron network with different stimulus histories for: (a)-(b) an uncoupled network; (c)-(d) a coupled network with a positive definite coupling matrix $\mathbf{Q}$; (e)-(f) a coupled network with a non-positive definite coupling matrix $\mathbf{Q}$.

3.5 Coupled spiking network with non-positive definite Q

As illustrated in Figure 8, a coupled spiking network can function as a memory element, when $\mathbf{Q}$ is a non-positive definite matrix and

$$\tau_i(t) = \phi\Big(h(t) * \sum_{j=1}^{M} Q_{ij}\Psi(v_j(t))\Big) \tag{42}$$

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 Figures 8(a), (c) and (e), and a different stimulus 'Stim 1b' precedes 'Stim 2' in Figures 8(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 Figures 8(a) and (b), independent of the history. For a coupled network with a positive definite coupling matrix $\mathbf{Q}$, 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 Figures 8(c) and (d). For a non-positive definite coupling matrix $\mathbf{Q}$ 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 Figure 8(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 [36]. Various works have studied associative memories using networks of spiking neuron models having different degrees of abstraction and architectural complexities [37, 38]. Here, we demonstrate using an associative memory network of 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

$$Q_{ij} = \frac{1}{k}\sum_{s=1}^{S} t_i^s t_j^s, \tag{43}$$

where k is a scaling factor and $\boldsymbol{t}^s \in [0,1]^M,\ s = 1, ..., S$, 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

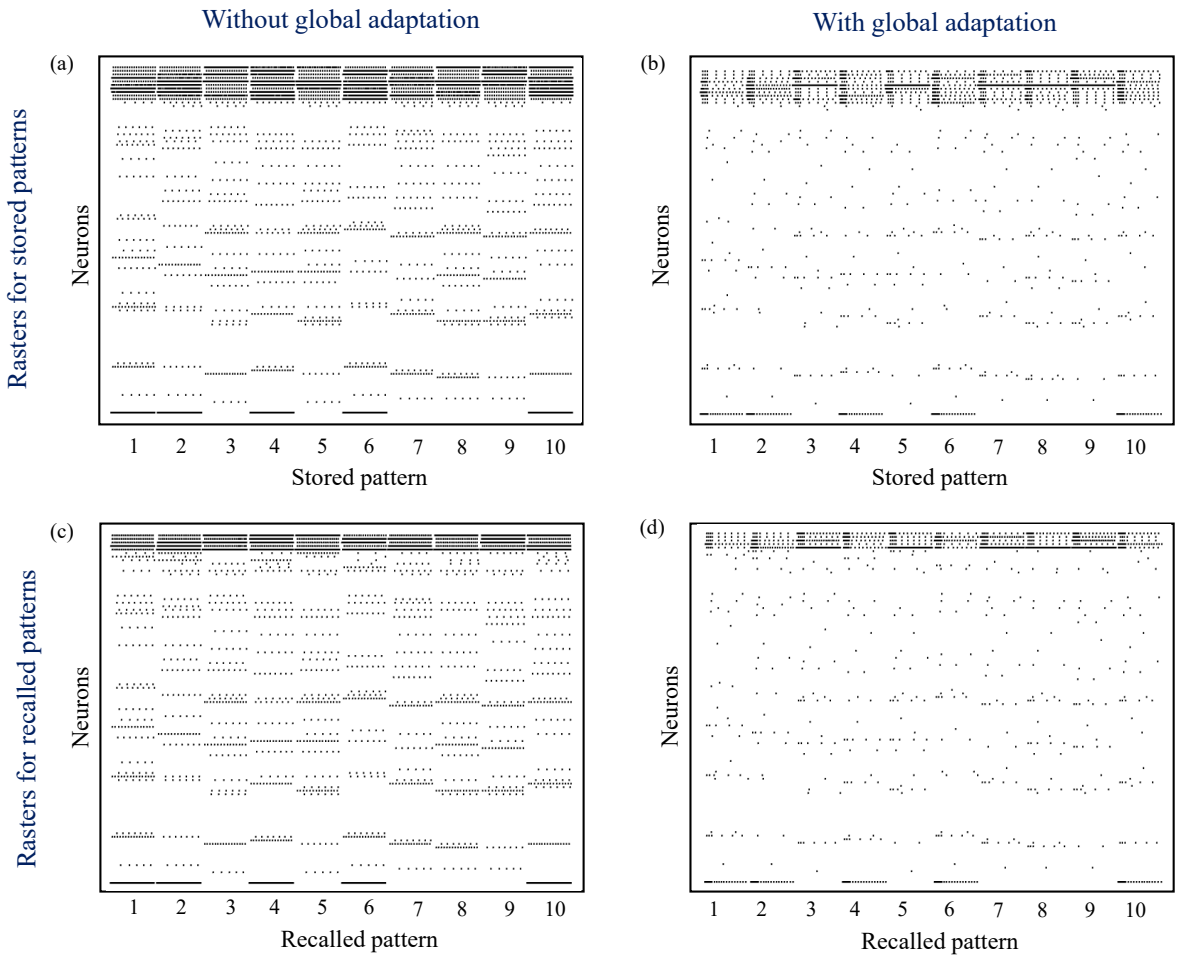

Figure 9: (a)-(b) Spike rasters for the 10 stored patterns without and with global adaptation respectively; (c)-(d) Spike rasters for the 10 recall cases without and with global adaptation respectively.

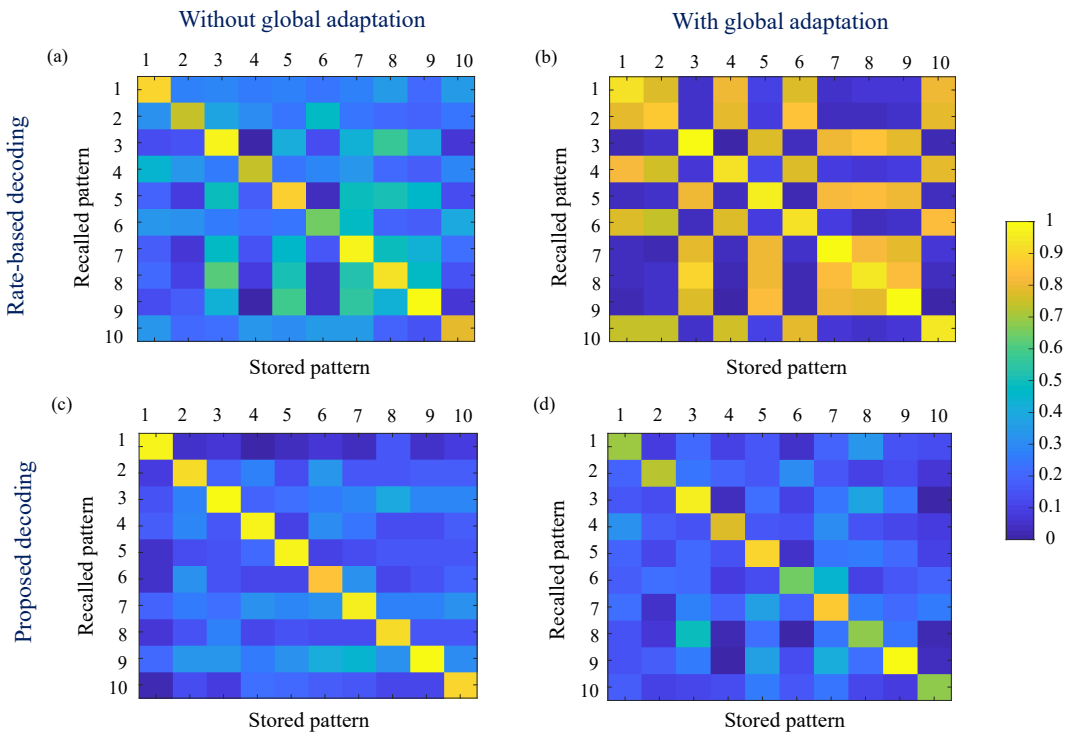

Figure 10: (a)-(b) Similarity matrices between storage and recall with a rate-based decoding metric; (c)-(d) Similarity matrices with a decoding metric that also includes spike-times and changes in mean firing rate.

is recorded. Instead of determining 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. Figures 9 (a) and (b) show the raster plots for the stored patterns without and with global adaptation respectively when $S = 10$; and Figures 9 (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 Figures 10(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 patterns, we get clean recalls in both cases as shown in Figures 10(c) and (d). The decoding vector for the n-th time-bin in this case is given by

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

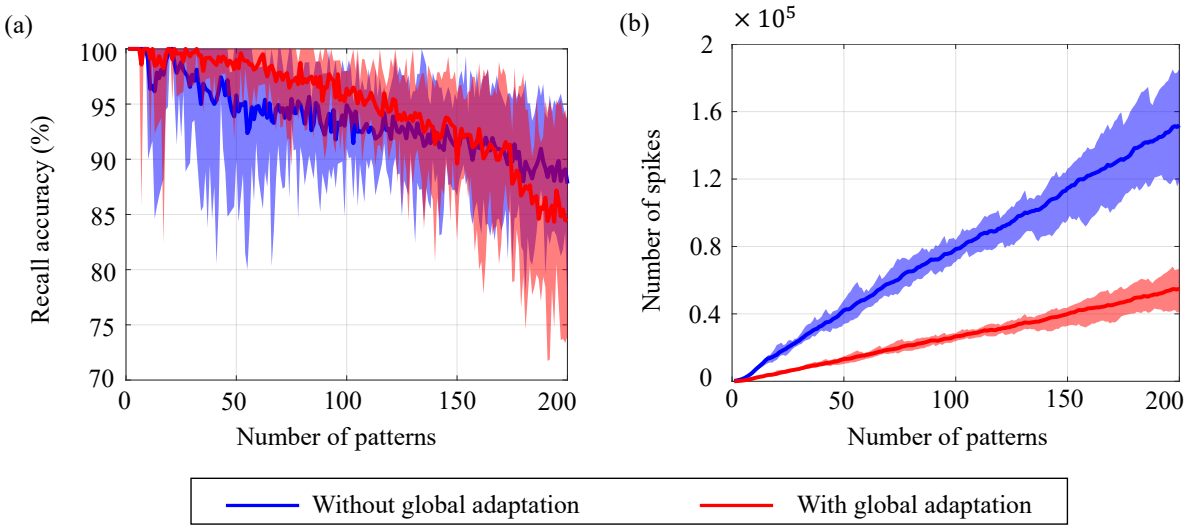

Figure 11: Ensemble plots showing (a) mean recall accuracy and (b) mean number of spikes as memory load increases for the network, in the absence as well as presence of global adaptation. The range of values across the ensemble is shown by the shaded area.

where $\boldsymbol{r}_n$, $\Delta\boldsymbol{t}_{\Psi n}$ and $\Delta\boldsymbol{r}_n$ are the vectors of mean firing rates, mean inter-spike intervals and changes in the mean firing rates for the n-th bin for the entire network respectively. The mean inter-spike interval 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. Note that the inter-spike interval computed for one time-bin may be different from $(1/r)$, particularly for low firing rates, and hence encodes useful information. The similarity metric between the u-th stored pattern and the v-th recall pattern is given by

$$s_{u,v} = 1 - \text{dist}_{u,v}, \tag{45}$$

where $\text{dist}_{u,v}$ is the mean Euclidean distance between the two decoding vectors over the total number of time-bins, normalized between $[0, 1]$.

To estimate the capacity of the network, we calculate the mean recall accuracy over 10 trials for varying number of stored patterns, both with and without global adaptation. Figure 11(a) plots the mean recall accuracy for different number of patterns stored for the two cases, and Figure 11(b) plots the mean number of spikes for each storage. For each plot, the shaded region indicates the range of values across trials. As expected, the accuracy is 100% for lesser storage,but degrades with higher loading. However with global adaptation, the degradation is seen to be more graceful for a large range of storage with the decoding used in Figures 10(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.

Note that the recall accuracy using global adaptation deteriorates faster for > 175 patterns. The proposed decoding algorithm, which determines the recall accuracy, takes into account the mean spiking rates, inter-spike intervals and changes in spike rates. It is possible that as the number of spikes is reduced through the use of global adaption, the information encoded in first-order differences (inter-spike intervals or spike rates) may not be sufficient to encode information at high fidelity, resulting in the degradation in recall accuracy when the number of patterns increased. However, augmenting the decoding features with higher-order differences in inter-spike intervals or spike rates may lead to an improved performance for higher storage.

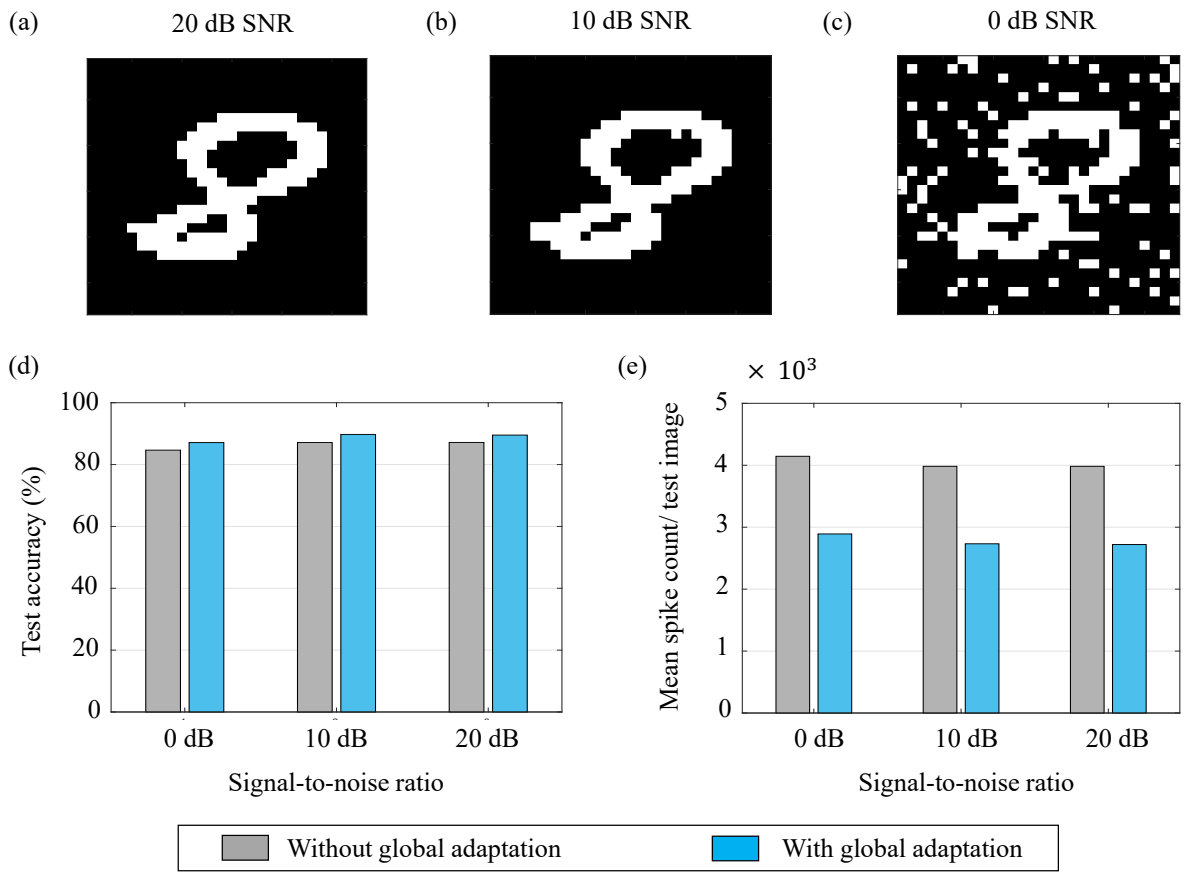

Figure 12: (a), (b) and (c) An example of a test image corrupted with additive white Gaussian noise at 20 dB, 10 dB and 0 dB SNR respectively; (d) and (e) Test accuracy and mean spike count/test image for different noise levels.

3.6.1 Classification of noisy MNIST images

Aside from pattern completion, associative networks are also commonly used for identifying patterns from their noisy counterparts. We use a similar associative memory network as above to classify images from the MNIST dataset which were corrupted with additive white Gaussian noise at different signal-to-noise ratios (SNRs), and which were, unlike in the previous case, unseen by the network before the recall phase. The network size in this case was $M = 784$, the number of pixels in each image, and the connectivity matrix was set using a separate, randomly selected subset of 5000 binary, thresholded images from the training dataset according to (43). Unseen images from the test dataset were corrupted at different SNRs and fed to the network after binary thresholding. Figures 12(a)-(c) shows an example of a test image at different SNRs after binary thresholding. As before, the non-zero pixels got a steady depolarizing input. A noisy test image was assigned to the class corresponding to the closest training image according to the similarity metric in (45).

The test accuracies and mean spike counts for a test image are plotted in Figures 12(d) and (e) respectively for different noise levels. We see that even for relatively high noise levels, the network has a robust classification performance. As before, a global adaptation based on the state of convergence of the network produces a slightly better performance with fewer spikes per test image.

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 proposed work is the first of its kind to treat the spike generation and transmission processes in a spiking network as an energy-minimization problem involving continuous-valued neural state variables like the membrane potential. 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.

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 parameters. The paper and the tool illustrate how dynamical and spiking responses of neurons can be derived directly from a network objective or energy functional of continuous-values neural variables. The general approach offers an elegant way to design neuromorphic machine learning algorithms by bridging the gap that currently exists between bottom-up models that can simulate biologically realistic neural dynamics but do not have a network-level representation, and top-down machine learning models that start with a network loss function, but reduce the problem to the model of a non-spiking neuron with static nonlinearities.

In this regard, machine learning models are primarily developed with the objective of minimizing the error in inference by designing a loss function that captures dependencies among variables, for example, features and class labels. Learning in this case, as pointed out in [12], consists of adapting weights in order to associate low energies (losses) to observed configurations of variables, and high energies (losses) to unobserved ones. The non-differentiable nature of spiking dynamics makes it difficult to formulate loss functions 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 [39, 40], formulating loss functions that penalize the difference between actual and desired spike-times [41, 42], or approximating the derivatives of spike signals through various means [43, 44, 45]. 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. Moreover, it would allow us to incorporate and experiment with biologically relevant neural dynamics that could have significant performance and energy benefits.

4.1 Relation with other neural networks and spiking neuron models

The network energy functional bears similarity with the Ising Hamiltonians used in Hopfield networks [46], Boltzmann machines [47] or spin-glass models [48], but contains an additional integral term $\int \Psi(.)dv$ as in continuous-time Hopfield networks with graded neurons [49]. However, unlike in continuous-time Hopfield networks where $\Psi^{-1}(.)$ is assumed to be a saturation/squashing function of a rate-based representation, the role of $\Psi(.)$ in the proposed model is to implement a barrier or a penalty, such that the neural responses can produce spiking dynamics. This enables us to obtain neural responses at the level of individual spikes instead of average rate-based responses; and allows for a more fine-grained control over the spiking responses of the network. The saturation (squashing) function, on the other hand, is implemented by the bound constraints on the Growth Transform updates, and hence the network is not limited to choosing a specific form of saturation non-linearity (e.g. sigmoid).

The energy-based formulation described in Section 2.1 could also admit other novel inter-

pretations. For instance, for the form of $\Psi(.)$ considered in 18, the barrier function can be rewritten as $\int_{-\infty}^{v_i} \Psi(v)dv = \Psi_i v_i$, where $\Psi_i = 0\ A$ if $v_i \leq 0\ V$ and $\Psi_i = I_\Psi\ A$ if $v_i > 0\ V$. For a continuous-time implementation (discrete-time step that is sufficiently small), $v_i(t)$ will be reset as soon as it reaches the threshold ($0\ V$), and will not exceed the threshold. In this case, we can write $\Psi_i(t) \geq 0$ and $\Psi_i(t)v_i(t) = 0\ \ \forall t$. This is equivalent to Karush-Kuhn-Tucker (KKT) conditions. Thus the spike events $\Psi_i(t),\ \ i = 1, ..., M$, act as the KKT multipliers corresponding to the M inequality constraints $v_i(t) \leq 0,\ \ i = 1, ..., M$ [50], encoding the sensitivity of the i-th neuron to the constraint.

Also, if we consider the spike response as a displacement current, we can write

$$o_i = C_{out}\frac{dv_i}{dt}, \tag{46}$$

where C_{out} is the membrane capacitance. Note that o_i is the analog spike response current output and is different from $\Psi(v_i)$, which is the binary spike event. Then the membrane potential for the continuous-time Growth Transform neuron model in Table 3 can be rewritten as

$$v_i(t) = \frac{1}{C_{out}}\int_{-\infty}^{t} o_i(t')dt'. \tag{47}$$

Thus according to this interpretation, the communication between neurons takes place using current waveforms, similar to integrate-and-fire models, and the current waveforms can be integrated at the post-synaptic neuron to recover the membrane potential. Note that the remapping between $\mathbf{W}$ and $\mathbf{Q}$ (described in Section 2.1) would still hold, since we are transmitting analog spike current waveforms, and not post-synaptic current waveforms such as exponentially decaying functions, α-functions or simplified current pulses (digital bits) used in integrate-and-fire models [51, 52].

4.2 Implication of remapping on neuromorphic architectures

In the proposed neuron model, we abstracted out the essential aspects of spike generation and transmission that can replicate neural dynamics, and remapped synaptic interactions to an energy-based framework. As a result of the remapping procedure, the coupling matrix $\mathbf{Q}$ in our proposed model is proportional to the inverse of the synaptic weight matrix $\mathbf{W}$. This paves the way for developing novel neuromorphic learning algorithms in the $\mathbf{Q}$-domain that involves sparse local analog connectivity, but which actually translates to fully-connected non-sparse global connectivity in the $\mathbf{W}$-domain. Thus, adapting one synaptic connection in the $\mathbf{Q}$-domain, in this case, will be equivalent to adapting multiple synapses in the $\mathbf{W}$-domain. Learning in the $\mathbf{Q}$-domain will be a topic for future research.

4.3 Benefits of decoupling neurodynamical parameters

A key advantage of the proposed framework is that it enables the decoupling of the three neurodynamical parameters - network solution, spike shapes and transient dynamics. Thus while the solution to the energy functional is determined by the coupling matrix $\mathbf{Q}$ and the stimulus vector $\mathbf{b}$, independent control of the modulation function allows users to program the trajectory to the solution, which could be determined by an optimization process that is different from optimizing the energy functional. Some examples of these alternate objectives could be:

- A hybrid spiking network comprising neurons of different types (tonic spiking, bursting, non-spiking, etc.), as illustrated in Section 3.1. The network would still converge to the

same solution, but the spiking dynamics across the network could be exploited to influence factors such as speed, energy efficiency and noise-sensitivity of information processing.

- Optimization of some auxiliary network parameter, e.g., the total spiking activity. A related example (although not optimized w.r.t. any objective function) was illustrated in Section 3.6 for a simple associative network. In this example, the network recalled the same set of patterns and classified MNIST images using two different time-evolutions of the modulation function corresponding to the presence and absence of global adaptation. In this case, it had the benefit of using fewer spikes to achieve better recall when a modified decoding metric was used.

- Effects of neurotransmitters and metabolic factors that have been known to affect the properties, activity and functional connectivity of populations of neurons. These factors endow the same network with the flexibility to generate different output patterns and produce different behaviors for the same stimulus [53, 54].

- Effects of diffusion processes or glial processes, that have been known to modulate response properties and synaptic transmission in neurons, influencing information processing and learning in the brain [55, 56].

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] 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.

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

[10] 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.

[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] H. Soula, G. Beslon, and O. Mazet, "Spontaneous dynamics of asymmetric random recurrent spiking neural networks," *Neural Computation*, vol. 18, no. 1, pp. 60–79, 2006.

[17] B. Cessac, "A discrete time neural network model with spiking neurons: Ii: Dynamics with noise," *Journal of mathematical biology*, vol. 62, no. 6, pp. 863–900, 2011.

[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] 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.

[21] 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.

[22] 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.

[23] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1063–1070, 2004.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

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

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

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

[39] 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.

[40] 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.

[41] 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.

[42] 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.

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

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

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

[46] 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.

[47] 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.

[48] 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.

[49] J. Šíma and P. Orponen, "Continuous-time symmetric hopfield nets are computationally universal," *Neural Computation*, vol. 15, no. 3, pp. 693–733, 2003.

[50] A. W. Tucker and H. Kuhn, "Nonlinear programming," in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics nd Probability, Univ. of California Press*, 1951, pp. 481–492.

[51] 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.

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

[53] M. Hasselmo, "Neuromodulation in mammalian nervous systems," *Handbook of brain theory and neural networks*, 2002.

[54] D. A. McCormick and M. P. Nusbaum, "Editorial overview: neuromodulation: tuning the properties of neurons, networks and behavior," *Current opinion in neurobiology*, vol. 29, p. iv, 2014.

[55] R. D. Fields, A. Araque, H. Johansen-Berg, S.-S. Lim, G. Lynch, K.-A. Nave, M. Nedergaard, R. Perez, T. Sejnowski, and H. Wake, "Glial biology in learning and cognition," *The neuroscientist*, vol. 20, no. 5, pp. 426–431, 2014.

[56] L. E. Clarke and B. A. Barres, "Emerging roles of astrocytes in neural circuit development," *Nature Reviews Neuroscience*, vol. 14, no. 5, pp. 311–321, 2013.

[57] 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.

Appendix A

We can rewrite (1) for a sequence $\big(v_{i,n-N+1}, v_{i,n-N+2}, ..., v_{i,n}\big)$ of size N to obtain

$$\begin{aligned}
v_{i,n-N+2} &= v_{i,n-N+1} + (\gamma-1)v_{i,n-N+1} + \sum_{j=1}^{M} W_{ij}\Psi(v_{j,n-N+1}) + y_{i,n-N+1} \\
v_{i,n-N+3} &= v_{i,n-N+2} + (\gamma-1)v_{i,n-N+2} + \sum_{j=1}^{M} W_{ij}\Psi(v_{j,n-N+2}) + y_{i,n-N+2} \\
&. \\
&. \\
v_{i,n+1} &= v_{i,n} + (\gamma-1)v_{i,n} + \sum_{j=1}^{M} W_{ij}\Psi(v_{j,n}) + y_{i,n}
\end{aligned}$$

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

$$(1-\gamma)\mathbb{E}_N\Big(v_i[n]\Big) + \frac{1}{N}\Big(v_{i,n+1} - v_{i,n-N+1}\Big) = \sum_{j=1}^{M} W_{ij}\mathbb{E}_N\Big(\Psi_j[n]\Big) + \mathbb{E}_N\Big(y_i[n]\Big), \tag{A.1}$$

where $\mathbb{E}_N\Big(\Psi_j[n]\Big) = \frac{1}{N}\sum\limits_{p=n-N+1}^{n}\Psi(v_{j,p})$. Since the neural responses are assumed to be bounded at all times, as $N \to \infty$, the second term in (A.1) approaches zero, so that we can rewrite (A.1) as

$$(1-\gamma)\bar{v}_i[n] = \sum_{j=1}^{M} W_{ij}\bar{\Psi}_j[n] + \bar{y}_i[n]. \tag{A.2}$$

Appendix B

Proof of Proposition I: We can decompose the scalar variable $v_{i,n}$ as $v_{i,n} = v_{i,n}^{+} - v_{i,n}^{-}$, where $v_{i,n}^{+}, v_{i,n}^{-} \geq 0$. The following additional constraint

$$v_{i,n}^{+} + v_{i,n}^{-} = v_c, \;\; i = 1, ..., M, \tag{B.1}$$

imposed on the variables $v_{i,n}^+$ and $v_{i,n}^-$ would then always ensure that (10) is satisfied. Then we can write

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

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 a Lipschitz continuous objective function under similar equality constraints [57, 20].

Baum-Eagon Growth Transformations: Baum-Eagon Growth Transforms [20] are a class of fixed-point algorithms for iteratively optimizing a Lipschitz continuous objective function $\mathcal{H}(\{v_{ik}\})$ that is constrained over a domain $\mathcal{D}$ defined by

$$\mathcal{D} = \big\{v_{ik} : v_{ik} \geq 0 \ \text{ and } \ \sum_{k=1}^{q_i} v_{ik} = v_c\big\}, \tag{B.3}$$

where $q_1, q_2, ..., q_M$ is a set of non-negative integers and M is a positive integer which denotes the number of linear constraints in $\mathcal{D}$. For a Lipschitz continuous cost function $\mathcal{H}(\{v_{ik}\})$ where $v_{ik} \in \mathcal{D},\ \ i = 1, ..., M$, there exists a growth transformation $\sigma(.)$ such that

$$\mathcal{H}\big(\{\sigma(v_{ik})\}\big) \leq \mathcal{H}\big(\{v_{ik}\}\big)\ \ \forall v_{ik} \in \mathcal{D}. \tag{B.4}$$

This transformation takes the following form

$$\sigma(v_{ik,n}) = v_{ik,n+1} \leftarrow v_c \frac{G_{ik}(\boldsymbol{v}_n, \lambda)}{G_i(\boldsymbol{v}_n, \lambda)},\ \ i = 1, ..., M, \tag{B.5}$$

where

$$G_{ik}(\boldsymbol{v}_n, \lambda) \ = \ v_{ik,n}\big(-\frac{\partial \mathcal{H}}{\partial v_{ik,n}} + \lambda\big), \ \text{ and} \tag{B.6}$$

$$G_i(\boldsymbol{v}_n, \lambda) \ = \ \sum_{k=1}^{q_i} G_{ik}(\boldsymbol{v}_n, \lambda). \tag{B.7}$$

An admissible value for the constant λ is such that for any $v_{ik,n} \in \mathcal{D}$, $G_{ik}(\boldsymbol{v}_n, \lambda) \geq 0$ and $G_i(\boldsymbol{v}_n, \lambda) > 0$ [57].

Growth Transform neuron model updates. Assuming that the partial derivatives for the network energy functional in (8) are bounded, the optimization problem for the proposed model is equivalent to the one in (B.2), being constrained on a domain equivalent to $\mathcal{D}$, where $q_1 = q_2 = ... = q_M = 2$, and $v_i^+ + v_i^- = v_c,\ \ i = 1, ..., M$ are the M linear constraints. Considering $v_i^+ = v_{i1}$ and $v_i^- = v_{i2}$, we can rewrite the update equations in (B.5) in terms of the new optimization variables corresponding to (B.2) to obtain the following discrete-time updates

$$v_{i,n+1}^+ \ = \ v_c \frac{v_{i,n}^+}{\mu_{i,n}}(-\frac{\partial \mathcal{H}}{\partial v_{i,n}^+} + \lambda) \tag{B.8}$$

$$v_{i,n+1}^- \ = \ v_c \frac{v_{i,n}^-}{\mu_{i,n}}(-\frac{\partial \mathcal{H}}{\partial v_{i,n}^-} + \lambda), \tag{B.9}$$

where

$$\mu_{i,n} = v_{i,n}^+(-\frac{\partial \mathcal{H}}{\partial v_{i,n}^+} + \lambda) + v_{i,n}^-(-\frac{\partial \mathcal{H}}{\partial v_{i,n}^-} + \lambda) \tag{B.10}$$

is a normalization factor that ensures $v_{i,n+1}^+ + v_{i,n+1}^- = v_c$. Here, λ (which has the unit of current)

is admissible iff $\lambda > |\frac{\partial \mathcal{H}}{\partial v_{i,n}}|\ \forall i, n$, which will ensure $G_i^+(\boldsymbol{v}_n, \lambda) > 0\ W$ and $G_i^-(\boldsymbol{v}_n, \lambda) > 0\ W\ \forall i$. From (B.8) and (B.9), using $v_{i,n} = v_{i,n}^+ - v_{i,n}^-$ and (B.1), we can easily show

$$v_{i,n+1} = v_c \frac{-\frac{\partial \mathcal{H}}{\partial v_{i,n}} v_c + \lambda v_{i,n}}{-\frac{\partial \mathcal{H}}{\partial v_{i,n}} v_{i,n} + \lambda v_c}\ , \tag{B.11}$$

where we have used the relation $\frac{\partial \mathcal{H}}{\partial v_i} = \frac{\partial \mathcal{H}}{\partial v_i^+} = -\frac{\partial \mathcal{H}}{\partial v_i^-}$. This dynamical system model, derived from the Growth Transform updates outlined in (B.8) and (B.9), ensures that (11) holds, with equality being satisfied iff $v_{i,n}$ is a critical point of $\mathcal{H}$.

Rearranging the terms in (B.11), we get

$$(v_c^2 - v_{i,n} v_{i,n+1}) \frac{\partial \mathcal{H}}{\partial v_{i,n}} + \lambda v_c (v_{i,n+1} - v_{i,n}) = 0. \tag{B.12}$$

Rewriting B.12 for a sequence of time-indices $p = n - N + 1, n - N + 2, ..., n$, of size N, we get

$$\begin{aligned}
(v_c^2 - v_{i,n-N+1} v_{i,n-N+2}) \frac{\partial \mathcal{H}}{\partial v_{i,n-N+1}} + \lambda v_c (v_{i,n-N+2} - v_{i,n-N+1}) &= 0 \\
(v_c^2 - v_{i,n-N+2} v_{i,n-N+3}) \frac{\partial \mathcal{H}}{\partial v_{i,n-N+2}} + \lambda v_c (v_{i,n-N+3} - v_{i,n-N+2}) &= 0 \\
&\vdots \\
(v_c^2 - v_{i,n} v_{i,n+1}) \frac{\partial \mathcal{H}}{\partial v_{i,n}} + \lambda v_c (v_{i,n+1} - v_{i,n}) &= 0
\end{aligned}$$

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

$$\mathbb{E}_N\Big(z_i[n]\Big) + \frac{1}{N} \lambda v_c (v_{i,n+1} - v_{i,n-N+1}) = 0 \tag{B.13}$$

where $z_{i,n} = (v_c^2 - v_{i,n} v_{i,n+1}) \frac{\partial \mathcal{H}}{\partial v_{i,n}}$. As $N \to \infty$, since $v_{i,n}$ are bounded $\forall i, n$, we have for the n-th time-window

$$\lim_{N \to \infty} \Big(\mathbb{E}_N\Big(z_i[n]\Big) \Big) \to 0. \tag{B.14}$$

Appendix C

Asymptotic encoding for non-saturating GT neurons:, For neurons with responses $v_{i,n} > -v_c\ \forall n$ (which includes spiking neurons as well as non-spiking neurons that do not cross the threshold), we define $\alpha_{i,n} = (v_c^2 - v_{i,n} v_{i,n+1})$. Since $|v_{i,n}| < v_c\ \forall n$, $\alpha_{i,n} > 0\ \forall n$. We can sum the criterion (B.12) for time-steps $p = n - N + 1, n - N + 2, ..., n$, to write

$$\sum_{p=n-N+1}^{n} \alpha_{i,p} g_{i,p} = \lambda v_c (v_{i,n-N+1} - v_{i,n+1}), \tag{C.1}$$

where $g_{i,p} = \frac{\partial \mathcal{H}}{\partial v_{i,p}}$. For $p = n - N + 1, n - N + 2, ..., n$, we can decompose the instantaneous gradient term $g_{i,p}$ as follows

$$g_{i,p} = \mathbb{E}_N\big(g_i[n]\big) + \Delta g_{i,p}, \tag{C.2}$$

where $\Delta g_{i,p}$ is a zero-mean sequence such that $\mathbb{E}_N\big(\Delta g_i[n]\big) = 0$. Then combining (C.1) and (C.2),

$$\begin{aligned}
\sum_{p=n-N+1}^{n} \alpha_{i,p}\mathbb{E}_N\big(g_i[n]\big) + \sum_{p=n-N+1}^{n} \alpha_{i,p}\Delta g_{i,p} &= \lambda v_c(v_{i,n-N+1} - v_{i,n+1}) \\
\mathbb{E}_N\big(g_i[n]\big) \sum_{p=n-N+1}^{n} \alpha_{i,p} &= -\sum_{p=n-N+1}^{n} \alpha_{i,p}\Delta g_{i,p} + \lambda v_c(v_{i,n-N+1} - v_{i,n+1}) \\
\mathbb{E}_N\big(g_i[n]\big) &= -\frac{\sum_{p=n-N+1}^{n} \alpha_{i,p}\Delta g_{i,p}}{\sum_{p=n-N+1}^{n} \alpha_{i,p}} + \frac{\lambda v_c(v_{i,n-N+1} - v_{i,n})}{\sum_{p=n-N+1}^{n} \alpha_{i,p}}
\end{aligned}$$

Since $\alpha_{i,n} > 0 \ \forall n$, $\sum_{p=n-N+1}^{n} \alpha_{i,p} \to \infty$ as $N \to \infty$. Also, $|v_{i,n}| < v_c \ \forall n$ leads to

$$\lim_{N\to\infty}\Big(\mathbb{E}_N\big(g_i[n]\big)\Big) = \lim_{N\to\infty}\Bigg(-\frac{\sum_{p=n-N+1}^{n} \alpha_{i,p}\Delta g_{i,p}}{\sum_{p=n-N+1}^{n} \alpha_{i,p}}\Bigg) = 0. \tag{C.3}$$

The last result uses the non-pathological case that due to the bounded property of $\Delta g_{i,p}$, the sequence $\alpha_{i,p}\Delta g_{i,p}$ does not grow as fast as the denominator sequence $\alpha_{i,p}$.

Appendix D

Continuous-time Growth Transform dynamical system: In order to derive the complete dynamical system model for the Growth Transform neuron, we apply a useful property of Growth Transforms. The Growth Transform mapping $\sigma(.)$ homotopically minimizes the value of the cost function $\mathcal{H}$ [9] as shown below

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

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

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

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

From (B.11), (D.2) and (D.3), we have

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

We define $\tau_{i,n} = \Delta t(1/f_{i,n} - 1)$, where Δt is the time-increment in seconds between two time-steps as defined previously in (1). Then since $0 < f_{i,n} \leq 1$, we have $\tau_{i,n} \in [0,\infty)$ s. $\tau_{i,n}$ can be considered to be the time-constant for the i-th neuron at the n-th time-step, and the discrete-time

dynamical systems model in (D.4) can be written as

$$(\tau_{i,n} + \Delta t)\left(\frac{v_{i,n+1} - v_{i,n}}{\Delta t}\right) + v_{i,n} = v_c \frac{-\frac{\partial \mathcal{H}}{\partial v_{i,n}} v_c + \lambda v_{i,n}}{-\frac{\partial \mathcal{H}}{\partial v_{i,n}} v_{i,n} + \lambda v_c}. \tag{D.5}$$

Since the n-th time-step corresponds to time $t = n\Delta t$, $v_{i,n} \equiv v_i(n\Delta t)$ and (D.5) can be rewritten as

$$(\tau_i(t) + \Delta t)\left(\frac{v_i(t + \Delta t) - v_i(t)}{\Delta t}\right) + v_i(t) = v_c \frac{-\frac{\partial \mathcal{H}}{\partial v_i(t)} v_c + \lambda v_i(t)}{-\frac{\partial \mathcal{H}}{\partial v_i(t)} v_i(t) + \lambda v_c} \tag{D.6}$$

In the limiting case when $\Delta t \to 0$ s, this reduces to the following continuous-time dynamical system model [20]

$$\tau_i(t)\frac{dv_i(t)}{dt} + v_i(t) = v_c \frac{-\frac{\partial \mathcal{H}}{\partial v_i(t)} v_c + \lambda v_i(t)}{-\frac{\partial \mathcal{H}}{\partial v_i(t)} v_i(t) + \lambda v_c}. \tag{D.7}$$