Computation by Switching in Complex Networks of States

Fabio Schittler Neves and Marc Timme

*Network Dynamics Group, Max Planck Institute for Dynamics and Self-Organization, and Bernstein Center for Computational
Neuroscience, Göttingen 37073, Germany*

Complex networks of dynamically connected saddle states persistently emerge in a broad range of high-dimensional systems and may reliably encode inputs as specific switching trajectories. Their computational capabilities, however, are far from being understood. Here, we analyze how symmetry-breaking inhomogeneities naturally induce predictable persistent switching dynamics across such networks. We show that such systems are capable of computing arbitrary logic operations by entering into switching sequences in a controlled way. This dynamics thus offers a highly flexible new kind of computation based on switching along complex networks of states.

Can complex networks of states of a nonlinear dynamical system be used to solve computational tasks? If so, which systems are appropriate and how can computations be performed? A broad range of systems exhibit saddle states that are dynamically linked via heteroclinic connections to form complex networks [1–16]. These may be promising candidates for such computations because the dynamics close to heteroclinic networks is intrinsically robust, easily controllable, and provides a large number of state-changing options already for small systems [8,9,12,16]. A heteroclinic connection in a dynamical system is a distinguished trajectory that links two saddles in state space. It occurs if unstable directions of one saddle are contained in the stable manifold of a second. A sequence of such connections linking several saddles cyclically is called a heteroclinic cycle. If the dynamical system considered exhibits a certain symmetry, complex heteroclinic networks consisting of interconnected heteroclinic cycles emerge in a robust way.

Heteroclinic networks are of high current mathematical interest [1–5,15]. Simultaneously, their specific dynamical features—supporting repetitive switching close to the saddles—pose a promising challenge for the study of information encoding and computation, in particular, in artificial neural systems [6–14,16]. For instance, it becomes more and more clear how information may be encoded by systems with heteroclinic cycles [8,10,11,14,16]. Recent studies even provide insights about how external perturbations may be processed [12,15,16] and hints on how such systems may actually compute using switching among saddles [12]. Nevertheless, it is still far from understood whether and how nonlinear dynamical systems may perform generic computations exploiting their complex networks of states.

In this Letter, we introduce the concept of computation via dynamics close to complex networks of states. We show how symmetry-breaking external signals naturally induce cyclic switching dynamics close to what has been a heteroclinic cycle in a symmetric system (Fig. 1), and

how to exploit such dynamics to execute generic logic operations. We exemplify our results by analyzing networks of neuronlike elements. Already a small system of only $N = 5$ units may realize any (out of two) unary, any
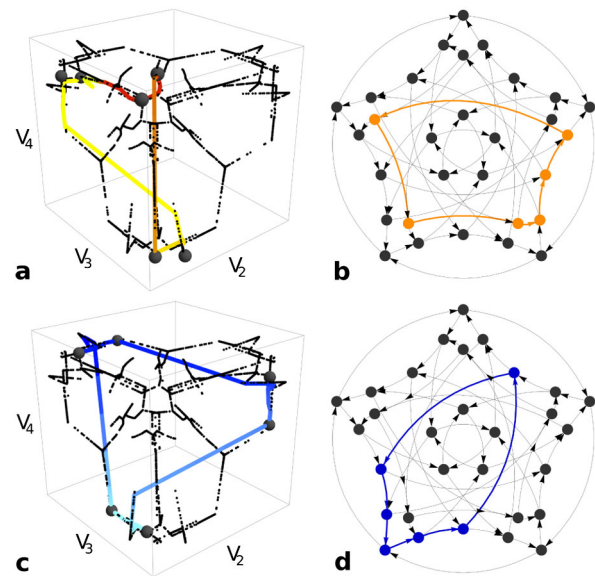

FIG. 1 (color online).   Selecting cyclic switching dynamics in complex networks of saddle states [29]. (a)–(d) Black: complex networks of states in the absence of inputs. Colors: trajectories in state space induced by specific symmetry-breaking currents, $\mathbf{\Delta} = (4, 3, 2, 1, 0) \times 10^{-4}$ in (a),(b) and $\mathbf{\Delta} = (1, 4, 3, 2, 0) \times 10^{-4}$ in (c),(d). (a),(c) State space representation. Whenever oscillator $i = 1$ is reset, the potentials of oscillators $i \in \{2, 3, 4\}$ are plotted as a three-dimensional trajectory; the potential of oscillator $i = 5$ is given by a color gradient from zero (light color) to one (dark color). Small black dots represent all possible switching trajectories (generated by low-noise-induced switchings). Gray spheres represent the vicinity of approached saddle states. (b),(d) Abstract representation of the same network of 30 saddles. Colored lines indicate the specific cyclic sequence of saddles approached by the trajectories shown in (a) and (b).

(out of 16) binary, and any (out of 256) ternary function. The underlying mechanism is rooted in a sequential sorting process performed at each saddle state. We discuss the advantageous scaling of computational capabilities with network size and illustrate it by an example of a system of $N = 100$ units. In addition, the signal-induced switching (i) appears as a generic feature of symmetry breaking in systems with heteroclinic connections, (ii) yields intrinsic self-corrections due to cyclic repetitions, and (iii) is robust to distractions and simultaneously sensitive to distinct external signals. Given these features, this computational paradigm provides a natural and flexible approach for performing logic operations in a variety of systems [1–16].

Consider a system of N oscillators interacting via exchanging delayed pulses [7,9,17]. The potential variable $V_i(t)$ of each oscillator $i \in \{1, 2, \ldots, N\}$ at time t changes according to

$$\frac{dV_i}{dt} = f(V_i) + W_i(t) + \Delta_i, \qquad (1)$$

where $f(V_i)$ is a monotonic concave-down differentiable function, Δ_i is a symmetry-breaking current, and

$$W_i(t) = \sum_{\substack{j=1 \\ j \neq i}}^{N} \sum_{m \in \mathbb{Z}} \epsilon \delta(t - \tau - t_{j,m}) \qquad (2)$$

are the pulse interactions received by oscillator i from others j. Here, ϵ is the homogeneous coupling strength and $\delta(.)$ represents the delta distribution. At times $t_{j,m}$, oscillator j reaches its threshold value at unity for the mth time, and $V_j(t_{j,m}^-) \geq 1$ is reset to $V_j(t_{j,m}) := 0$ and emits a pulse that is received by all other oscillators after a delay $\tau > 0$ [9].

For homogeneous currents, $\Delta_i \equiv \Delta_j$ for all i, j, such systems robustly exhibit a closed network of saddle states. Typically, such a saddle is a cluster state with several groups of synchronized units (clusters). These states are related by permutation symmetry and specifically linked via heteroclinic connections. For instance, for an analytically tractable system of $N = 5$ oscillators, the units synchronize into a three-cluster configuration with symmetry $S_2 \times S_2 \times S_1$, where S_n is the permutation group of n elements. Each of these $5!/(2!2!1!) = 30$ states are saddles with identical stability properties. Upon small perturbations, each cluster may either dissolve or resynchronize [16]. For the system considered, the dynamics is unstable to perturbations desynchronizing one specific two-unit cluster, but stable to perturbations to the other two-unit cluster or the single-unit cluster (S_1). In order to simplify the notation, we substitute the real values $\{V_i\}$ by cluster labels $V_i \in \{a, b, c\}$ such that a saddle state, e.g., $\mathbf{V}^* = (a, a, b, b, c)$, is unstable against perturbations that desynchronize the units labeled a but stable against all other perturbations.

A small instantaneous perturbation signal applied to the system residing in such a saddle induces a transition to one specific other saddle [16]. For instance, given initial state $\mathbf{V}^*$, an arbitrary small perturbation $\boldsymbol{\delta}$ with $\delta_1 > \delta_2$ yields the transition

$$(a, a, b, b, c) + (\delta_1, \delta_2, \delta_3, \delta_4, \delta_5) \rightarrow (c, b, a, a, b), \quad (3)$$

which is independent of δ_3, δ_4, and δ_5 because there is only one unstable direction (in which the a cluster desynchronizes). Therefore, if the a cluster is desynchronized, the system switches to another saddle; if a perturbation does not desynchronize the a cluster, the system stays at the same saddle. Thus, any perturbation desynchronizing the a cluster has the same effect as a single-unit perturbation, e.g.,

$$(a, a, b, b, c) + (\delta, 0, 0, 0, 0) \rightarrow (c, b, a, a, b) \quad (4)$$

for $\delta = \delta_1 - \delta_2 > 0$. For $\delta < 0$, the transition is $(a + \delta, a, b, b, c) \rightarrow (b, c, a, a, b)$. Due to the permutation symmetry of the cluster states, the same single-unit transitions

$$a + \delta \rightarrow c, \qquad a + 0 \rightarrow b,$$
$$\text{both } b \rightarrow a \quad \text{and} \quad c \rightarrow b \qquad (5)$$

are induced from any of the 30 initial configurations if one unit of the a cluster is driven more strongly, $a + \delta$, than the other, $a + 0$.

The permutations of Eq. (4) thus yield the complete set of possible transitions and define a complex network of 30 dynamically connected states [Figs. 1(a)–1(d), network indicated in black]. This network is closed in the sense that a sufficiently small instantaneous generic perturbation to the system anywhere on the cycle induces switching but keeps the trajectory inside the network. Here, the detailed form of the external symmetry-breaking perturbation is not relevant because any quantity added to the right-hand side of Eq. (1) cannot be distinguished from others after temporal integration. Throughout this work, we take a temporally constant symmetry-breaking vector specified by nonidentical external currents Δ.

To understand the emergence of switching cycles in more detail, we characterize each cycle through the sequence of saddles it approaches. Given an initial state (c, b, a, a, b) and an external current, Δ, with

$$\Delta_1 > \Delta_2 > \ldots > \Delta_5, \qquad (6)$$

yields the cyclic sequence of saddles

$$(c, b, a, a, b) \rightarrow (b, a, c, b, a) \rightarrow (a, c, b, a, b)$$
$$\uparrow \qquad\qquad\qquad\qquad \downarrow$$
$$(a, c, b, b, a) \leftarrow (b, a, c, a, b) \leftarrow (c, b, a, b, a)$$

according to Eqs. (3)–(6). This cyclic switching process thus consists of a sequential comparison of the two

potential values in the *a* cluster generating a specific cyclic sequence [Fig. 1 (colored trajectories)]. The transients prior to such cycles given any initial condition in the basin of attraction of the heteroclinic network are typically short (see Supplemental Material [18]).

According to the state sequence in Eq. (7), the values V_1, V_2, and V_3 are never compared among themselves. Moreover, V_4 is never compared to V_5. Therefore, the switching dynamics reveals no information about the relative magnitude of asymmetries within these two groups, but distinguishes the three strongest from the two weakest inputs.

How can these features be exploited computationally? Near each saddle, the switching dynamics amounts to comparing the effective frequencies of the two oscillators in the *a* cluster. The dynamics along one switching cycle of six saddles thus provides a dynamical classification of five-dimensional real-valued vector inputs into $\binom{5}{3} = 10$ discrete classes. In other words, the real-valued asymmetry is internally perceived (classified) as a discrete binary vector $\mathbf{r} = (r_1, r_2, r_3, r_4, r_5)$, where $r_i \in \{0, 1\}$ and $\sum_i r_i = 3$. This input classification provides basic forms of computation. First, it relates input signals to discrete internal representations independent of the 30 possible initial saddle states; and second, all classes, and thus the computation, are predictable by simple transitions rules, (4) and (5).

To demonstrate the computational capabilities of this complex network of 30 saddle states, we exploit the systematic switching dynamics to implement a logic computer. It has three contributions. First, the input, coded by an *N*-bit binary vector $\mathbf{R}$ that selects both the logic operation (last *m* bits) and defines the operand bits (first $N - m$ bits). Second, inputs $\mathbf{R}$ and a base current $\boldsymbol{\Delta}$ combine to an effective asymmetry vector

$$\boldsymbol{\Delta}_{\mathrm{eff}} = \boldsymbol{\Delta} + \boldsymbol{\omega} \circ \mathbf{R}, \qquad (8)$$

where $\boldsymbol{\omega}$ is a weight vector and $\circ$ means element-wise multiplication (see Supplemental Material for more details [18]). This induces a specific cyclic pattern (sequence of saddles) characterized by a vector $\mathbf{r}$. Third, the cyclic pattern is interpreted as a binary output (two states) according to the operation implemented to visualize the result in a binary representation.

For interpretation, we explicitly read out the cyclic pattern by detecting simultaneous events using a delay plasticity principle [19]. Here, the delays between the oscillator network and the readout are such that the pulses are perceived as simultaneous events by the readout when close to a given saddle. In principle, any device that is capable of detecting synchronized spikes may be used as a readout [20]. Our approach uses integrate-and-fire neurons and thus the same units as the oscillator network. Such a unified setting, using the same unit types for the oscillatory system and its readout, may be advantageous when

implementing the computational paradigm in a hardware device (see below).

The system of $N = 5$ units exemplified above constitutes the smallest system exhibiting a robust heteroclinic network [16]. Already here, all unary, binary, or ternary logic operations can be straightforwardly implemented because the 10 classes always enable a one-to-one mapping from class to desired output. It can therefore realize both unary operations, i.e., the identity and the NOT operation on a single bit, by fixing $m = 4$ inputs; any one (out of $2^{(2^2)} = 16$) binary function by fixing $m = 3$ inputs and leaving the system with two inputs to be varied; and any arbitrary (out of $2^{(2^3)} = 256$) ternary function by fixing $m = 2$ inputs, leaving the system with three inputs to be varied. Figure 2 illustrates an XOR binary logic gate, explicitly showing that the temporally distributed processing, in particular, enables computations that are not linearly separable. All other abovementioned functions equally work (see Supplemental Material [18]).

Furthermore, it is possible to simultaneously realize more than one function in the same network (at fixed network parameters and fixed base asymmetry). Here, the fact that different operations often exhibit the same output (e.g., zero) for a given input lowers the number of cyclic sequences required. For example, with a careful choice of parameters $(\boldsymbol{\Delta}, \mathbf{R})$, three binary functions, such as AND, OR, and XOR, may be realized simultaneously (see Supplemental Material for details [18]).

The sorting processes described emerge independent of the exact realization of unit dynamics, form of coupling, or network size, because they rely only on the existence of robust heteroclinic networks of saddles. Qualitatively, the same computations can thus be performed by a wide variety of systems, not restricted to neuronlike systems [2,5,6,8,9,12]. We illustrate this in a system of $N = 100$ units with modified intrinsic dynamics [7,9] that equally
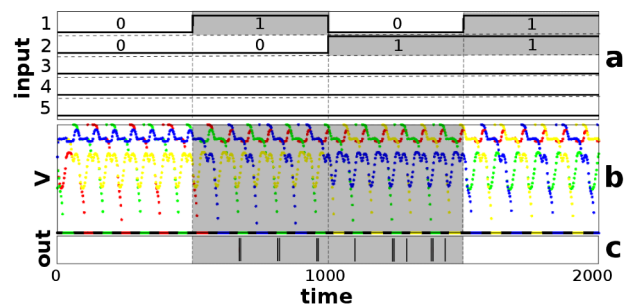

FIG. 2 (color online).   XOR binary logic gate via cyclic switching dynamics across networks of states. In (a)–(c), white background represents 0 (nonactivity) and gray 1 (activity). (a) Fixing the three inputs $R_{3,4,5} = (0, 0, 0)$ leaves two inputs to vary $R_{1,2} \in \{(0, 0), (1, 0), (0, 1), (1, 1)\}$. Each of the four input combinations yields (b) a specific limit cycle. The potentials of all oscillators are plotted whenever oscillator $i = 1$ is reset. (c) A single output unit decodes the limit cycles by spiking or not spiking, thereby providing a binary output.
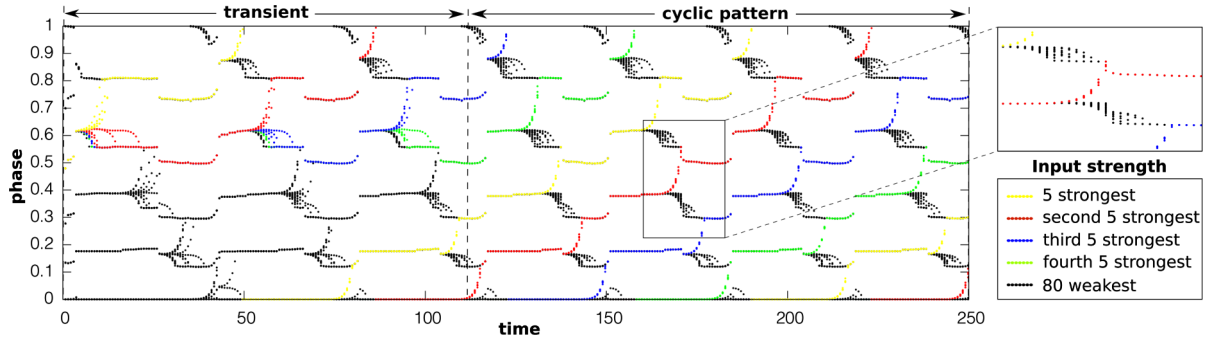
FIG. 3 (color online). One 20-winner-take-all computation via switching dynamics for $N = 100$; cluster symmetry $S_{21} \times S_{21} \times S_{21} \times S_{21} \times S_{16}$ [30]. The phase of all oscillators is plotted when oscillator $i = 1$ is reset. The phase is a monotonic function of the potential given by $\phi_i = (e^{bV_i} - 1)/(e^b - 1)$. Asymmetric external currents $\Delta = (\Delta_1, \ldots, \Delta_{100})$ induce specific limit cycles that serve as internal representations of the computational outcome. The initial condition is such that all 20 largest inputs are delivered to elements in the same cluster. Close to every saddle, the units in one specific, desynchronizing cluster that are subject to the five largest inputs leave that cluster. The resulting dynamics is such that four groups of five units that in total form the 20 largest inputs repeatedly leave their clusters when close to the saddle where it is unstable.

exhibits a heteroclinic network, but now with cluster symmetry $S_{21} \times S_{21} \times S_{21} \times S_{21} \times S_{16}$ (Fig. 3). A small asymmetric current generalizing Eq. (6) induces a specific limit cycle across a certain part of the complex network of $100!/([21!]^4 16!) \approx 6.5 \times 10^{65}$ saddles in 100-dimensional space. Again, sorting sequentially occurs at each saddle visited such that the sorting process now separates the units receiving the largest 20 inputs from the remaining 80 (Fig. 3). The real-valued asymmetry is internally perceived as a binary vector $\mathbf{r} = (r_1, \ldots, r_{100})$ where $r_i \in \{0, 1\}$ and $\sum_i r_i = 20$, thus yielding $\binom{100}{20} \approx 5.4 \times 10^{20}$ discrete classes. Therefore, the system of $N = 100$ units realizes a complex 20-winner-take-all computation in the sense that exactly those units that receive the 20 largest inputs repeatedly leave clusters.

For general $N \gg 1$, where the typical number of clusters is between two and about 10 [7] and the typical number of distinguishable switching elements $K = \alpha N$ is roughly 20% to 50% of N, the number of classes $M_C = \binom{N}{\alpha N}$ grows much faster than linearly with system size. For instance, if $\alpha = 1/2$, we have

$$M_C = \binom{N}{N/2} \approx \frac{2^{N+1/2}}{\sqrt{\pi N}} \qquad (9)$$

for large N using the Stirling approximation.

In summary, we have demonstrated that switching across complex networks of states that are robustly linked via heteroclinic connections offers a natural way to solve computational tasks by dynamical systems. In previous works on coding and dynamic response properties of oscillatory circuits [8,12], individual saddles serve as conceptual coding elements. Such works also addressed the dynamics of biological neural circuits via modeling [8], and there are interesting experimental hints about functionally useful switching dynamics [21]. In the concept of the computation introduced above, we now represent inputs as cyclic sequences among saddle states [14] and exploit the

advantages coming from this reinterpretation, e.g., the insensitivity of the dynamical coding to the choice of the initial (saddle) state or cycle and to transient distractors.

Computations exploiting complex networks of states join many interesting features present in other approaches of neural computation. For instance, echo and liquid state approaches [22,23] also exploit the high-dimensional dynamics and their intrinsic instabilities to provide a fixed representation system with flexible readout options; stable attractor dynamics in recurrent neural networks [24,25] intrinsically provide features such as controllability and predictability; and certain dynamics of formal spiking neurons [26] use neuronlike features to provide natural (but not robust) 1-winner-takes-all dynamics. The concept introduced in this Letter now exhibits several of the advantageous properties simultaneously and comes with the additional feature of self-correction via cyclic repetition. It exploits instabilities in high-dimensional dynamics, yet is predictable (via simple transition rules) and thus controllable; it is intrinsically robust, yet provides a huge number of internal representations; and finally, it offers robust execution of a general k-winner-take-all computation.

In practical applications, such dynamics can be implemented as networks of different oscillator types [8,9,12] because only the feature of appropriate sorting at the saddle is required. One promising option that seems in direct experimental reach may be systems of coupled lasers [27,28].

Taken together, we introduced the concept of computation exploiting complex networks of states that emerge in a wide range of systems. The combinatorial scaling of the number of possible computations with system size and the independence from the specific system makes this approach accessible for future theoretical studies and experimental implementations. Switching across heteroclinic networks of saddles thus provides a promising way to perform generic computations by complex networks of

states. Understanding the computational capabilities of such persistent switching dynamics in more detail will help to efficiently design more complex computations; for example, the cyclic spatiotemporal pattern itself (instead of a binary output) could serve as an input to a downstream device.

[1] J. Guckenheimer and P. Holmes, Math. Proc. Cambridge Philos. Soc. **103**, 189 (1988).

[2] C. M. Postlethwaite and J. H. P. Dawes, Nonlinearity **18**, 1477 (2005).

[3] J. H. P. Dawes and T. L. Tsai, Phys. Rev. E **74**, 055201 (2006).

[4] H. Broer, K. Efstathiou, and E. Subramanian, Nonlinearity **21**, 13 (2008); **21**, 1385 (2008).

[5] H. Kori and Y. Kuramoto, Phys. Rev. E **63**, 046214 (2001); H. Kori, ibid. **68**, 021919 (2003).

[6] D. Hansel, G. Mato, and C. Meunier, Phys. Rev. E **48**, 3470 (1993).

[7] U. Ernst, K. Pawelzik, and T. Geisel, Phys. Rev. Lett. **74**, 1570 (1995); Phys. Rev. E **57**, 2150 (1998).

[8] M. Rabinovich, A. Volkovskii, P. Lecanda, R. Huerta, H. D. I. Abarbanel, and G. Laurent, Phys. Rev. Lett. **87**, 068102 (2001).

[9] M. Timme, F. Wolf, and T. Geisel, Phys. Rev. Lett. **89**, 154105 (2002); Chaos **13**, 377 (2003).

[10] P. Ashwin, G. Orosz, and S. Townley, SIAM J. Appl. Dyn. Syst. **6**, 728 (2007).

[11] G. Laurent, M. Stopfer, R. W. Friedrich, M. I. Rabinovich, A. Volkovskii, and H. D. I. Abarbanel, Annu. Rev. Neurosci. **24**, 263 (2001).

[12] P. Ashwin and J. Borresen, Phys. Rev. E **70**, 026203 (2004); Phys. Lett. A **347**, 208 (2005).

[13] C. Kirst and M. Timme, Phys. Rev. E **78**, 065201 (2008).

[14] C. Bick and M. I. Rabinovich, Phys. Rev. Lett. **103**, 218101 (2009).

[15] G. Orosz, P. Ashwin, J. Wordsworth, and S. Townley, Proc. Appl. Math. Mech. **7**, 1030703 (2007).

[16] F. S. Neves and M. Timme, J. Phys. A **42**, 345103 (2009).

[17] R. E. Mirollo and S. H. Strogatz, SIAM J. Appl. Math. **50**, 1645 (1990).

[18] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.109.018701 for further details about the computational options provided by complex heteroclinic networks of states.

[19] C. W. Eurich, K. Pawelzik, U. Ernst, J. D. Cowan, and J. G. Milton, Phys. Rev. Lett. **82**, 1594 (1999).

[20] R. Gütig and H. Sompolinsky, Nat. Neurosci. **9**, 420 (2006).

[21] G. Laurent, Nat. Rev. Neurosci. **3**, 884 (2002).

[22] W. Maass, T. Natschläger, and H. Markram, Neural Comput. **14**, 2531 (2002).

[23] H. Jaeger and H. Haas, Science **304**, 78 (2004).

[24] J. J. Hopfield, Proc. Natl. Acad. Sci. U.S.A. **79**, 2554 (1982).

[25] J. Hertz, A. Krogh, and R. G. Palmer, Introduction to the Theory of Neural Computation (Westview, Boulder, CO, 1991).

[26] D. Z. Jin and H. S. Seung, Phys. Rev. E **65**, 051922 (2002).

[27] A. Amann, A. Pokrovskiy, S. Osborne, and S. O'Brien, J. Phys. Conf. Ser. **138**, 012001 (2008).

[28] V. Flukert and E. Schöll, New J. Phys. **14**, 033039 (2012).

[29] Leaky integrate-and-fire oscillators [7,9]: $f(V) = I - \gamma V$, $\tau = 0.49$, $\epsilon = 0.025$, $I = 1.04$, $\gamma = 1$, $\mathbf{\Delta} = (4, 3, 2, 1, 0) \times 10^{-4}$, and $\omega = (2.5, 2.5, 2.5, 5, 5) \times 10^{-4}$.

[30] Mirollo-Strogatz oscillators [17]: $f(V) = (e^b - 1)e^{-bV}/b$, $\tau = 0.15$, $\epsilon = 0.2$, $b = 3$, all $\Delta_i \approx 10^{-5}$, and $\Delta_i > \Delta_j$ for all $i \le 20$ and $j > 20$.