Editors' Suggestion Featured in Physics

Smooth Exact Gradient Descent Learning in Spiking Neural Networks

Christian Klos[*] and Raoul-Martin Memmesheimer[†]
Neural Network Dynamics and Computation, Institute of Genetics, University of Bonn, 53115 Bonn, Germany

(Received 19 December 2023; accepted 22 November 2024; published 13 January 2025)

Gradient descent prevails in artificial neural network training, but seems inept for spiking neural networks as small parameter changes can cause sudden, disruptive appearances and disappearances of spikes. Here, we demonstrate exact gradient descent based on continuously changing spiking dynamics. These are generated by neuron models whose spikes vanish and appear at the end of a trial, where it cannot influence subsequent dynamics. This also enables gradient-based spike addition and removal. We illustrate our scheme with various tasks and setups, including recurrent and deep, initially silent networks.

DOI: 10.1103/PhysRevLett.134.027301

Introduction—Biological neurons communicate via short electrical impulses called spikes [1]. Besides their overall rate of occurrence, the precise timing of single spikes often carries salient information [2–5]. Taking into account spikes is therefore essential for the modeling and the subsequent understanding of biological neural networks [1,6]. To build appropriate spiking network models, powerful and well-interpretable learning algorithms are needed. They are further required for neuromorphic computing, an aspiring field that develops spiking artificial neural hardware to apply them in machine learning. It aims to exploit properties of spikes such as event-based, parallel operation (neurons only need to be updated when they send or receive spikes) and the temporal and spatial (i.e., in terms of interacting neurons) sparsity of communication to achieve tasks with unprecedented energy efficiency and speed [7–9].

The prevalent approach for learning in nonspiking neural network models is to perform gradient descent on a loss function [10,11]. Importantly, during such learning the representations change continuously and in a predictable manner as the networks are compositions of functions that are continuous in the network parameters. The transfer of gradient descent learning to spiking networks is, however, problematic due to the all-or-none character of spikes: the appearance or disappearance of spikes is not predictable from gradients computed for nearby parameter values. This is because the gradient only accounts for changes in spike timing of those spikes present when it is computed. Thus, a systematic addition or removal of spikes via exact gradient descent is seemingly not possible. This can, for example, lead to permanently silent, so-called dead neurons [12,13] and to diverging gradients [14]. Further, the network dynamics after a spike appearance or disappearance may change in a disruptive manner [15–18]. This can result in discontinuous changes of the representations, which are given by the spike times, and of the loss during learning.

Nevertheless, there are two popular approaches for learning in spiking neural networks based on gradient descent. The first approach, surrogate gradient descent, assumes binned time and replaces the binary activation function with a continuous-valued surrogate for the computation of the gradient [19]. It thus sacrifices the crucial advantage of event-based processing and learning only from spikes and necessitates the computation of state variables in each time step as well as their storage [20] (but see [21]). Furthermore, the computed surrogate gradient is only an approximation of the true gradient. The second approach, spike-based gradient descent, computes the exact gradient of the loss by considering the times of existing spikes as functions of the learnable parameters [12,22]. It allows for event-based processing but relies on *ad hoc* measures to deal with spike appearances and disappearances and gradient divergence, in particular to avoid dead neurons [23–27].

Here, we show that disruptive appearances and disappearances of spikes can be avoided. Consequently, all network spike times vary continuously and in some network models even smoothly, i.e., continuously differentiably, with the network parameters. This allows us to perform nondisruptive, exact gradient descent learning, including, as we show, the systematic addition or removal of spikes.

Neuron model—The most frequently employed neuron models when learning spiking networks are variants of the leaky integrate-and-fire (LIF) neuron [14,18–24,26,28,60,61]. LIF neurons, however, suffer from the aforementioned disruptive spike appearance and disappearance. For example, spikes can appear in the middle of a trial due to a continuous, arbitrarily small change of an input weight or time [Figs. 1(a) and 1(b)]. Here and in the following, a trial refers to an individual run of an experiment with finite duration.

[*] Contact author: cklos@uni-bonn.de

[†] Contact author: rm.memmesheimer@uni-bonn.de

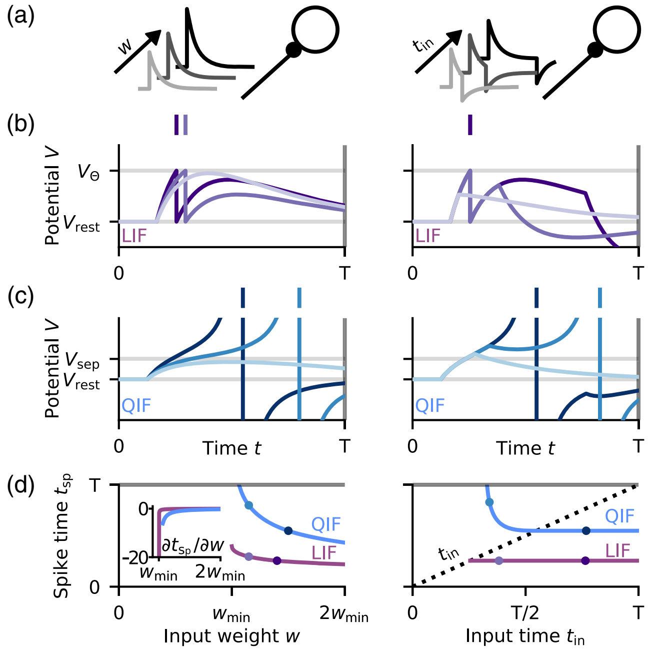

FIG. 1. Disruptive and nondisruptive appearance of spikes. (a),(b),(d) Spike times of the LIF neuron can appear disruptively in the middle of a trial. (a),(c),(d) Spike times of the QIF neuron only appear nondisruptively at the trial end and otherwise change continuously with changed parameters. Left column: a neuron receives a single input, whose weight is increased (traces with increasing saturation). Right column: a neuron receives an excitatory as well as an inhibitory input whose arrival is moved to larger times. (a) Setup (gray, different input currents). (b) LIF neuron membrane potentials [purple traces, saturation corresponding to (a); V_{rest} and V_Θ, resting and threshold potential; T, trial duration] and spikes (top, ticks). (c) Like (b) for the QIF neuron (V_{sep}, separatrix potential). (d) Times of the first output spike as function of the changed parameter; dots correspond to equally colored spikes in (b),(c). Left: w_{min}, weight at which the spike appears at finite time for the LIF neuron and at infinity for the QIF neuron. Inset: spike time gradient, divergent for the LIF neuron.

We therefore consider instead another important standard spiking neuron model, a quadratic integrate-and-fire (QIF) neuron [6,62,63]. Its membrane potential dynamics are governed by $\dot{V} = V(V-1) + I$, where I consists of temporally extended, exponentially decaying synaptic input currents, $\tau_{\mathrm{s}}\dot{I} = -I + \tau_{\mathrm{s}}\sum_i w_i \sum_{t_i}\delta(t-t_i)$. Here, τ_{s} is the synaptic time constant, measured in multiples of the membrane time constant $\tau_{\mathrm{m}} = 1$, and i indexes the presynaptic neurons, which spike at times t_i and have a synaptic weight w_i. In contrast to the LIF neuron, where $\dot{V}$ decays linearly with V, the QIF neuron explicitly incorporates the fact that in biological neurons the membrane potential further increases due to a self-amplification mechanism once it is large enough. As this generates spike upstrokes, the QIF neuron may be considered as the simplest truly spiking neuron model [63]. The voltage self-amplification is so strong that the voltage actually reaches infinity in finite time. One can define the time when this happens as the time of the spike, reset, and onset of synaptic transmission. We adopt this and henceforth call positive infinity the threshold of the QIF neuron for simplicity. For sufficiently negative voltage, the voltage increases strongly as well. The neuron can thus be reset to negative infinity, from where it quickly recovers. For LIF neurons, one needs to define finite threshold and reset potentials.

Nondisruptive appearance and disappearance of spikes and smooth spike timing—In the QIF neuron, spike times only appear and disappear at the trial end; otherwise they change smoothly with the network parameters. Importantly, this kind of spike appearance and disappearance is nondisruptive since the are no more spiking dynamics after the trial end that could be affected.

The mechanism underlying this feature can be intuitively understood: the voltage slope $\dot{V}$ at the threshold is infinitely large. If there is a small change, for example, in an input weight (Fig. 1, left column, blue curves), V and $\dot{V}$ will still be large close to where the spike has previously been. Therefore a spike will still be generated, only a bit earlier or later, unless it crosses the trial end. This is in contrast to the LIF neuron, where $\dot{V}$ at the threshold can tend to zero and a spike can therefore abruptly appear or disappear, accompanied by a diverging gradient (Fig. 1, left column, purple curves). A similar mechanism applies if there are changes in an input time as in Fig. 1, right column: an inhibitory input is moved backward in time until it crosses the time of an output spike generated by a sole, previous excitatory input [t_{in} crosses t_{sp} in Fig. 1(d) right]. In the QIF neuron, V and $\dot{V}$ are infinitely large at this point, such that the additional inhibitory input is negligible compared to the intrinsic drive. Thus there is no abrupt change in spike timing. In contrast, in the LIF neuron the inhibitory input induces a downward slope in the potential also if it is at the threshold. The spike induced by the excitatory input alone therefore suddenly appears once the inhibitory input arrives later.

In Supplemental Material, Sec. II [28], we prove the smoothness of the spike times and their nondisruptive appearance and disappearance in the general case with multiple inputs and output spikes.

Pseudodynamics and pseudospikes—The nondisruptive disappearance of spikes allows spike-based gradient descent to remove them in a controlled manner by shifting them past the trial end. In contrast, the gradient contains no information about spike appearances at the trial end, precluding the systematic addition of spikes. Being able to add spikes is, however, important because a neuron may initially or at some point during learning spike insufficiently often for the task or even be completely silent.

To solve this problem, we appropriately extend the ordinary dynamics by what we call pseudodynamics.

Concretely, we propose two types of pseudodynamics. In the first type, which we use in our applications, the neurons continue to evolve as QIF neurons, but with an added constant, suprathreshold drive, until they have spiked sufficiently often for the task [28]. We call the additional spikes pseudospikes. They only affect the pseudodynamics of postsynaptic neurons by controlling the value of the added drive. This ensures generically nonzero gradients. The continued evolution as a QIF neuron ensures continuity and mostly smoothness of the spike times, even if a spike transitions from a pseudospike to an ordinary one. In Supplemental Material, Sec. IB [28], we suggest a second approach where the spike times remain completely smooth.

Both types of pseudospike times have several useful properties: (i) they depend continuously and mostly smoothly on the network parameters, also when the pseudospikes cross the trial end to turn into ordinary spikes. (ii) If the voltage at the trial end increases, the pseudospike times decrease, intuitively because the neuron is already closer to spike. (iii) Pseudospikes affect postsynaptic pseudospikes but not ordinary ones. (iv) The pseudospikes interact such that the components of the gradient in multilayer networks are generically nonzero also if neurons are inactive during the actual trial duration. (v) The pseudospike times are computable in closed form.

Similar pseudospike time functions can be found for other neuron and synapse models with continuous spike times such as QIF neurons with infinitesimally short synaptic currents that generate voltage jumps [28].

Gradient descent learning—In the following, we apply spike-based gradient descent learning on the neural network models with continuous spike times identified above. We choose single neuron models with a closed-form solution between spikes and for the time of an upcoming spike. The former enables and the latter simplifies the use of efficient event-based simulations and modern automatic differentiation libraries [64,65] (The code to reproduce these results is publicly available [66].).

Interestingly, such solutions exist for the QIF neuron with temporally extended, exponentially decaying synaptic input currents if $\tau_s = \tau_m/2$ [28]. This is compatible with often assumed biologically plausible values, for example with a membrane time constant about 10 ms and a synaptic time constant about 5 ms [1,6]. In the examples in this Letter, we therefore use these values.

In the last of our three applications we employ oscillating QIF neurons with infinitesimally short input currents. Between spikes, they evolve with a constant rate of change in an appropriate phase representation [6,62,63,67], which further simplifies the event-based simulations. While their spike times are continuous, they are not smooth, as the derivative with respect to the time or weight of an input spike time jumps if it crosses another one.

Single neuron learning—As a first illustration of our scheme, we learn spike times of a single QIF neuron [Fig. 2(a); see [28] for details on models and tasks]. Initially it does not spike at all during the trial [Fig. 2(b), left]. We apply spike-based gradient descent to minimize the quadratic difference between two target and the first two output spike times (which may also be pseudospike times). The neuron is set to initially generate two pseudospikes, one for each target spike time. While not necessary in the displayed task, superfluous (pseudo)spikes can be included into the loss function with target behind the trial end to induce their removal if they enter the trial.

The use of pseudospikes allows one to activate the initially silent neuron [Fig. 2(c), gray background]. In doing so, the pseudospike times transition smoothly into ordinary spike times [Fig. 2(c), white background]. They are then shifted further until they lie precisely at the desired position on the time axis [Fig. 2(b), right]. The spike times change smoothly [Fig. 2(c)] and the loss gradient is continuous [Fig. 2(d)]. The example illustrates that our scheme allows one to learn precisely timed spikes of a single neuron in a smooth fashion and even if the neuron is initially silent.

Learning a recurrent neural network—Next, we consider the training of a recurrent neural network (RNN),

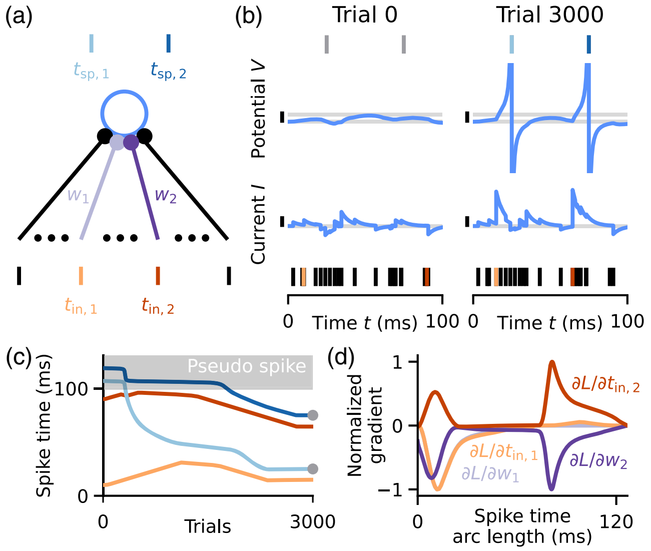

FIG. 2. Smooth gradient descent learning in a QIF neuron. (a) Weights (purple) and times (orange) of two inputs are learned to adjust the first two output spike times (blue). (b) Left: before learning, the neuron does not spike (gray ticks, target spike times; horizontal gray lines, V_{sep}, V_{rest}, or zero input current; black bars, potential or current difference of 1; orange, black ticks, learned, other input spikes). Right: after learning, the neuron spikes at the desired times (blue ticks cover gray ticks). (c) During learning, the [pseudo (gray area)] spike times change smoothly [colors as in (a); gray circles, target spike times]. (d) The components of the gradient of the loss function L change continuously during learning ($\partial L/\partial w_1$ mostly covered by $\partial L/\partial t_{in,1}$). Learning progress is displayed as a function of the arc length of the output spike time trajectories since the start of learning [28].

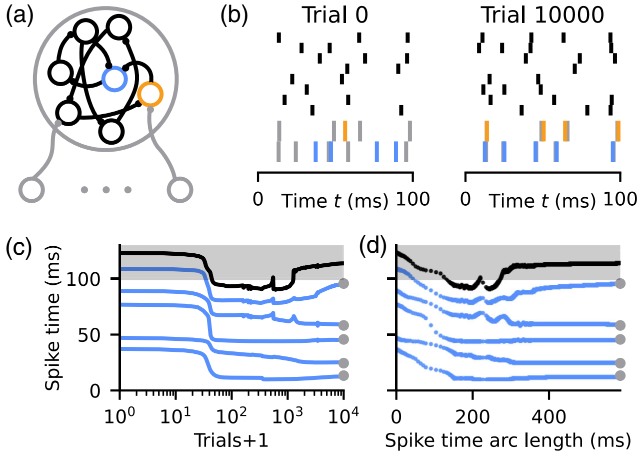

FIG. 3. Learning precise spikes in a RNN. (a) Network schematic. Neurons receive in each trial the same spikes from external input neurons (gray). Spike times of the first two network neurons are learned (blue and orange). (b) Spikes of network neurons before (left) and after (right) learning (colored ticks, spikes of first two neurons; gray ticks, target times). (c) Spike time trajectories of the first neuron during learning. Desired spikes (blue traces) shift toward their target times (gray circles). The first superfluous spike (black trace) is pushed out of the trial. (Gray area indicates pseudospikes.) (d) Same as (c) but the spike times are shown as a function of the arc length of the output spike time trajectories [28], which demonstrates their continuity, despite the occurrence of large gradients [cf. the steplike change in (c)].

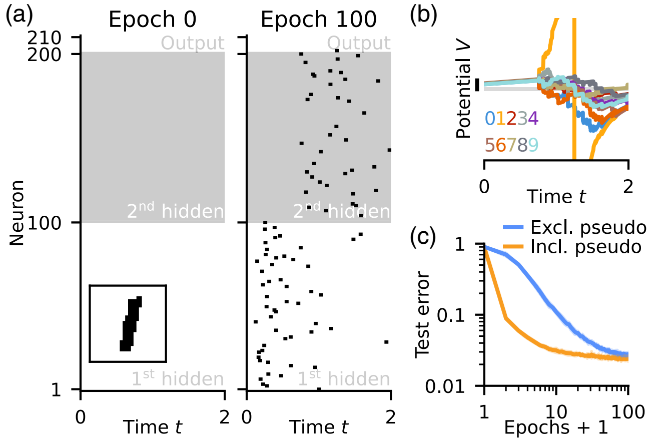

FIG. 4. MNIST task. (a) Spike raster plot of the three-layer network. Left: silent neurons before learning. Inset: example input also used on the right and in (b). Right: sparse spiking after learning. (b) Voltage dynamics of the output neurons after learning (horizontal gray line, V_{rest}; black bar, potential difference of 1). (c) Classification error dynamics. Utilizing pseudospikes also during testing (orange) generates smaller test errors in early training (solid lines indicate mean and shaded areas standard deviation over ten network instances).

where spike time changes have a global impact. It can be useful for the reconstruction of cortical networks [18,68,69]. We consider a fully connected RNN of ten QIF neurons with external inputs and learn the spike times of two network neurons by updating the recurrent weights and initial conditions [Fig. 3(a)]. In contrast to the learning of all network spikes [18,70], this does not reduce to independently finding a mapping from given input spike times to output spike times for each neuron.

Our scheme is successful also in this scenario and the spike times are precisely learned [Fig. 3(b)]. As in the previous example, the spike times of the first neuron change continuously during learning without discrete jumps [Figs. 3(c) and 3(d)]. Because of large gradients, which are typical for all kinds of RNNs [71], some changes are jumplike for the second neuron [28]. The underlying continuity becomes clear when restricting the maximal spike time change per step using adjustable update step sizes [28]. Hence, this example illustrates the applicability of our scheme to recurrent networks.

Standard machine learning task—Finally, we apply our scheme to the classification of hand-written single-digit numbers from the MNIST dataset, which is a widely used benchmark in neuromorphic computing (e.g., [20,24,61]).

We employ a three-layer feedforward network consisting of oscillatory QIF neurons with infinitesimally short input currents. For each input pixel, there is a corresponding input neuron, which spikes once at the beginning of the trial if the binarized pixel intensity is 1 and otherwise remains silent. The index of the neuron in the output layer that spikes first is the model prediction [72].

To demonstrate that our scheme allows one to solve the dead neuron problem even if neurons in multiple layers are silent, we randomly initialize network parameters such that there are initially basically no ordinary spikes [Fig. 4(a), left]. Yet, the pseudospike time-dependent, imposed interaction between the neurons allows to backpropagate errors. Hence, the hidden and output neurons are activated [Figs. 4(a) right, (b)]. Finally basically all hidden neurons spike before the first output spike for some input image [28], indicating that they contribute to inference. Still activity is sparse. The final accuracy of 97.3% when only considering ordinary output spikes is comparable to previous results with similar setups [23–25,73]. If we also allow pseudospikes during testing (which only affects trials without ordinary output spikes), the accuracy does not change much. The minimal error level is, however, reached faster [Fig. 4(c)]. Thus, our scheme achieves competitive performance in a neuromorphic benchmark task even if almost no neurons are initially active.

Discussion—We have shown that there are neural networks with spike times that vary continuously or even smoothly with network parameters; ordinary spikes only appear and disappear at the trial end and can be extended to pseudospikes. The networks allow one to learn the timings of an arbitrary number of spikes in a continuous fashion with a spike-based gradient.

Perhaps surprisingly, the networks may consist of rather simple, standard QIF neurons. These are widely used in theoretical neuroscience [6,63], also for supervised learning [68,74,75], and have been implemented in neuromorphic hardware [76,77]. However, the particularity that spikes only appear and disappear at the trial end has not been noticed and exploited. We expect also that further neuron models exhibit spikes with continuous timings if their voltage slope close to the threshold is guaranteed to be positive. This includes neuron models that generate spikes by reaching infinite voltage, such as hybrid leaky integrate-and-fire neurons with an attached, nonlinear spike generation mechanism [78], the Izhikevich neuron with minor modifications [63], the rapid theta neuron [79,80], the sine neuron [81], and the exponential integrate-and-fire neuron [6]. It further includes intrinsically oscillating LIF neurons and antileaky integrate-and-fire neurons [82], if the impact of synaptic input currents vanishes at their spike threshold. We also expect that synapses with continuous current rise will be feasible, as well as conductance-based synapses.

On the one hand, our scheme possesses the same advantages as other spike-based gradient descent approaches such as small memory and computational footprints and a clear interpretation as following the exact loss gradient. On the other hand, like standard machine learning schemes it produces no disruptive transitions during learning and no gradient divergences. This suggests a wide range of applications: when studying biological neural networks, our scheme may be used to learn neurobiologically relevant tasks, to benchmark biological learning, to investigate how the network dynamical solutions may work, and to reconstruct synaptic connectivity from experimentally (partially) observed spiking activity. Furthermore, it may be used to train networks in neuromorphic computing (see [28] for further discussion). It generally allows one to benchmark other learning rules whose underlying mechanisms are less transparent and to train and pretrain networks before converting to a desired neuron type that complicates learning.

The dynamics of spiking and nonspiking neural networks can be chaotic [82–86] and give rise to exploding gradients [10,28,71,87]. We therefore restricted our learning examples to at most ten multiples of the membrane time constant. This fits the length of various experimentally observed precisely timed spike patterns [2,88–92] and the fast processing of certain tasks in neuromorphic computing [20,23–25,73].

Our pseudospikes allow the gradient to "see" spikes before they appear and to thus add spikes in a systematic manner. Pseudospikes affect the pseudospikes of postsynaptic neurons and ultimately of the output neurons. This preserves the gradients of the ordinary spike times and solves, in particular, the dead neuron problem. In a somewhat related approach, silent output neurons were assumed to spike at the trial end [26,27]. Our pseudospikes, however, apply to all neurons and allow one to backpropagate errors through silent neurons. The resulting possibility of initializing an entire network with small weights may be important to induce desirable and biologically plausible features such as energy-efficient final connectivity and sparse spiking [7,93], sparse coding [94], and representation learning [95].

To conclude, the present study shows something that seemed fundamentally impossible [8]: despite the inherent discreteness of spikes, there can be exact nondisruptive, even smooth gradient descent learning in spiking neural networks, including the gradient-based removal and, after augmentation with pseudodynamics, also generation of spikes.

Acknowledgments—We thank Sven Goedeke for helpful comments on the manuscript and the German Federal Ministry of Education and Research (BMBF) for support via the Bernstein Network (Bernsteinpreis 2014, 01GQ1710).

[1] P. Dayan and L. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems* (MIT Press, Cambridge, MA, 2001).

[2] T. Gollisch and M. Meister, Rapid neural coding in the retina with relative spike latencies, Science **319**, 1108 (2008).

[3] J. Wolfe, A. R. Houweling, and M. Brecht, Sparse and powerful cortical spikes, Curr. Opin. Neurobiol. **20**, 306 (2010).

[4] H. P. Saal, X. Wang, and S. J. Bensmaia, Importance of spike timing in touch: An analogy with hearing?, Curr. Opin. Neurobiol. **40**, 142 (2016).

[5] S. J. Sober, S. Sponberg, I. Nemenman, and L. H. Ting, Millisecond spike timing codes for motor control, Trends Neurosci. **41**, 644 (2018).

[6] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal Dynamics—From Single Neurons to Networks and Models of Cognition* (Cambridge University Press, Cambridge, England, 2014).

[7] M. Pfeiffer and T. Pfeil, Deep learning with spiking neurons: Opportunities and challenges., Front. Neurosci. **12**, 774 (2018).

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

[9] C. D. Schuman, S. R. Kulkarni, M. Parsa, J. P. Mitchell, P. Date, and B. Kay, Opportunities for neuromorphic computing algorithms and applications, Nat. Comput. Sci. **2**, 10 (2022).

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016).

[11] N. Kriegeskorte and T. Golan, Neural network models and deep learning, Curr. Biol. **29**, R231 (2019).

[12] J. K. Eshraghian, M. Ward, E. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu,

Training spiking neural networks using lessons from deep learning, arXiv:2109.12894.

[13] A. Taherkhani, A. Belatreche, Y. Li, G. Cosma, L. P. Maguire, and T. McGinnity, A review of learning in biologically plausible spiking neural networks, Neural Netw. **122**, 253 (2020).

[14] O. Booij and H. tat Nguyen, A gradient descent rule for spiking neurons emitting multiple spikes, Inf. Proc. Lett. **95**, 552 (2005).

[15] C. van Vreeswijk and H. Sompolinsky, Chaotic balanced state in a model of cortical circuits, Neural Comput. **10**, 1321 (1998).

[16] S. Jahnke, R.-M. Memmesheimer, and M. Timme, Stable irregular dynamics in complex neural networks, Phys. Rev. Lett. **100**, 048102 (2008).

[17] M. Monteforte and F. Wolf, Dynamic flux tubes form reservoirs of stability in neuronal circuits, Phys. Rev. X **2**, 041007 (2012).

[18] R.-M. Memmesheimer, R. Rubin, B. Ölveczky, and H. Sompolinsky, Learning precisely timed spikes, Neuron **82**, 925 (2014).

[19] E. O. Neftci, H. Mostafa, and F. Zenke, Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks, IEEE Signal Process. Mag. **36**, 51 (2019).

[20] T. C. Wunderlich and C. Pehle, Event-based backpropagation can compute exact gradients for spiking neural networks, Sci. Rep. **11**, 12829 (2021).

[21] N. Perez-Nieves and D. F. M. Goodman, Sparse spiking gradient descent, in *Advances in Neural Information Processing Systems*, edited by A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (Curran Associates, Inc., New York, 2021).

[22] S. M. Bohte, J. N. Kok, and H. L. Poutré, Error-backpropagation in temporally encoded networks of spiking neurons, Neurocomputing **48**, 17 (2002).

[23] I. M. Comsa, K. Potempa, L. Versari, T. Fischbacher, A. Gesmundo, and J. Alakuijala, Temporal coding in spiking neural networks with alpha synaptic function, in *ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, Barcelona, 2020), pp. 8529–8533.

[24] J. Göltz, L. Kriener, A. Baumbach, S. Billaudelle, O. Breitwieser, B. Cramer, D. Dold, A. F. Kungl, W. Senn, J. Schemmel, K. Meier, and M. A. Petrovici, Fast and energy-efficient neuromorphic deep learning with first-spike times, Nat. Mach. Intell. **3**, 823 (2021).

[25] H. Mostafa, Supervised learning based on temporal coding in spiking neural networks, IEEE Trans. Neural Networks Learn. Syst. **29**, 3227 (2018).

[26] T. Nowotny, J. P. Turner, and J. C. Knight, Loss shaping enhances exact gradient learning with EventProp in spiking neural networks, arXiv:2212.01232.

[27] S. R. Kheradpisheh and T. Masquelier, Temporal backpropagation for spiking neural networks with one spike per neuron, International Journal of Neural Systems **30**, 2050027 (2020).

[28] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.134.027301 for further model and task details, smoothness and continuity proofs, analysis of gradient statistics, further simulation results, and further discussion, which includes Refs. [29–59]

[29] T. P. Vogels, K. Rajan, and L. Abbott, Neural network dynamics, Annu. Rev. Neurosci. **28**, 357 (2005).

[30] A. Burkitt, A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input, Biol. Cybern. **95**, 1 (2006).

[31] W. R. Inc., Mathematica, Version 13.2, Champaign, IL, 2023.

[32] E. Kamke, *Differentialgleichungen. Lösungsmethoden und Lösungen* (Teubner, Stuttgart, 1977).

[33] H. Tuckwell, *Introduction to Theoretical Neurobiology: Volume 1. Linear Cable Theory and Dendritic Structure* (Cambridge University Press, Cambridge, England, 1988).

[34] H. Tuckwell, *Introduction to Theoretical Neurobiology: Volume 2. Nonlinear and Stochastic Theories* (Cambridge University Press, Cambridge, England, 1988).

[35] N. Brunel, Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons, J. Comput. Neurosci. **8**, 183 (2000).

[36] R.-M. Memmesheimer, Quantitative prediction of intermittent high-frequency oscillations in neural networks with supralinear dendritic interactions, Proc. Natl. Acad. Sci. U.S.A. **107**, 11092 (2010).

[37] R. Mirollo and S. Strogatz, Synchronization of pulse coupled biological oscillators, SIAM J. Appl. Math. **50**, 1645 (1990).

[38] R.-M. Memmesheimer and M. Timme, Designing complex networks, Physica D (Amsterdam) **224**, 182 (2006).

[39] M. D'Haene, B. Schrauwen, J. Van Campenhout, and D. Stroobandt, Accelerating event-driven simulation of spiking neurons with multiple synaptic time constants, Neural Comput. **21**, 1068 (2009).

[40] R. Brette *et al.*, Simulation of networks of spiking neurons: A review of tools and strategies, J. Comput. Neurosci. **23**, 349 (2007).

[41] C. R. Harris *et al.*, Array programming with NumPy, Nature (London) **585**, 357 (2020).

[42] P. Virtanen *et al.* (SciPy 1.0 Contributors), SciPy 1.0: Fundamental algorithms for scientific computing in Python, Nat. Methods **17**, 261 (2020).

[43] A. Paszke *et al.*, PyTorch: An imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., New York, 2019), Vol. 32, pp. 8024–8035.

[44] I. Babuschkin *et al.*, *The DeepMind JAX Ecosystem* (2020), http://github.com/google-deepmind.

[45] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan, and I. Stoica, Ray: A distributed framework for emerging AI applications, in *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)* (USENIX Association, Carlsbad, CA, 2018), pp. 561–577.

[46] J. D. Hunter, Matplotlib: A 2D graphics environment, Comput. Sci. Eng. **9**, 90 (2007).

[47] M. A. Petroff, Accessible color sequences for data visualization, arXiv:2107.02270.

[48] W. Rudin, *Principles of Mathematical Analysis* (McGraw-Hill, New York, 1976).

[49] M. W. Hirsch and S. Smale, *Differential Equations, Dynamical Systems, and Linear Algebra*, Pure and Applied Mathematics Vol. 60 (Academic Press, San Diego [u.a.], 1974).

[50] G. Jetschke, *Mathematik der Selbstorganisation* (Harri Deutsch, Frankfurt am Main, 2009).

[51] V. I. Arnold, *Ordinary Differential Equations* (Springer, Berlin, 1992).

[52] H. Heuser, *Lehrbuch der Analysis. Teil 1* (Teubner-Verlag, Stuttgart, 1998).

[53] A. Stanojevic, S. Woźniak, G. Bellec, G. Cherubini, A. Pantazi, and W. Gerstner, An exact mapping from ReLU networks to spiking neural networks, Neural Netw. **168**, 74 (2023).

[54] A. Stanojevic, S. Woźniak, G. Bellec, G. Cherubini, A. Pantazi, and W. Gerstner, High-performance deep spiking neural networks with 0.3 spikes per neuron, arXiv: 2306.08744.

[55] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste, Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks, J. Mach. Learn. Res. **22**, 10882 (2021).

[56] S. I. Mirzadeh, K. Alizadeh-Vahid, S. Mehta, C. C. del Mundo, O. Tuzel, G. Samei, M. Rastegari, and M. Farajtabar, ReLU strikes back: Exploiting activation sparsity in large language models, in *The 12th International Conference on Learning Representations* (OpenReview.net, Vienna, 2024).

[57] D. V. Christensen *et al.*, 2022 roadmap on neuromorphic computing and engineering, Neuromorphic Comput. Eng. **2**, 022501 (2022).

[58] E. Nordlie, M.-O. Gewaltig, and H. E. Plesser, Towards reproducible descriptions of neuronal network models, PLoS Comput. Biol. **5**, 1 (2009).

[59] J. Zhuang, T. Tang, Y. Ding, S. Tatikonda, N. Dvornek, X. Papademetris, and J. Duncan, AdaBelief optimizer: Adapting stepsizes by the belief in observed gradients, in *NeurIPS 2020 Workshop: Deep Learning through Information Geometry* (Curran Associates, Inc., New York, 2020).

[60] F. Zenke and S. Ganguli, SuperSpike: Supervised learning in multilayer spiking neural networks, Neural Comput. **30**, 1514 (2018).

[61] B. Cramer, S. Billaudelle, S. Kanya, A. Leibfried, A. Grübl, V. Karasenko, C. Pehle, K. Schreiber, Y. Stradmann, J. Weis, J. Schemmel, and F. Zenke, Surrogate gradients for analog neuromorphic computing, Proc. Natl. Acad. Sci. U.S.A. **119**, e2109194119 (2022).

[62] P. E. Latham, B. J. Richmond, P. G. Nelson, and S. Nirenberg, Intrinsic dynamics in neuronal networks. I. Theory, J. Neurophysiol. **83**, 808 (2000).

[63] E. Izhikevich, *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting* (MIT Press, Cambridge, MA, 2007).

[64] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, JAX: Composable transformations of Python + NumPy programs (2018), http://github.com/google/jax.

[65] R. Engelken, SparseProp: Efficient event-based simulation and training of sparse recurrent spiking neural networks, in *Advances in Neural Information Processing Systems*, edited by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Curran Associates, Inc., New York, 2023), pp. 3638–3657, Vol. 36.

[66] C. Klos (2024), https://github.com/chklos/spikegd.

[67] B. Ermentrout and N. Kopell, Parabolic bursting in an excitable system coupled with a slow oscillation, SIAM J. Appl. Math. **46**, 233 (1986).

[68] C. M. Kim and C. C. Chow, Learning recurrent dynamics in spiking networks, eLife **7**, e37124 (2018).

[69] A. Das and I. R. Fiete, Systematic errors in connectivity inferred from activity in strongly recurrent networks, Nat. Neurosci. **23**, 1286 (2020).

[70] R.-M. Memmesheimer and M. Timme, Designing the dynamics of spiking neural networks, Phys. Rev. Lett. **97**, 188101 (2006).

[71] R. Pascanu, T. Mikolov, and Y. Bengio, On the difficulty of training recurrent neural networks, in *Proceedings of the 30th International Conference on Machine Learning, Proceedings of Machine Learning Research* Vol. 28, edited by S. Dasgupta and D. McAllester (PMLR, Atlanta, GA, 2013), pp. 1310–1318.

[72] S. Thorpe, A. Delorme, and R. Van Rullen, Spike-based strategies for rapid processing, Neural Netw. **14**, 715 (2001).

[73] Y. Sakemi, K. Morino, T. Morie, and K. Aihara, A supervised learning algorithm for multilayer spiking neural networks based on temporal coding toward energy-efficient VLSI processor design, IEEE Trans. Neural Networks Learn. Syst. **34**, 394 (2023).

[74] D. Huh and T. J. Sejnowski, Gradient descent for spiking neural networks, in *Advances in Neural Information Processing Systems 31*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates, Inc., New York, 2018), pp. 1439–1449.

[75] S. McKennoch, T. Voegtlin, and L. Bushnell, Spike-timing error backpropagation in theta neuron networks, Neural Comput. **21**, 9 (2009).

[76] E. Basham and D. Parent, An analog circuit implementation of a quadratic integrate and fire neuron, in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (IEEE, New York, 2009).

[77] E. J. Basham and D. W. Parent, A neuromorphic quadratic, integrate, and fire silicon neuron with adaptive gain, in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (IEEE, New York, 2018).

[78] M. Pospischil, Z. Piwkowska, T. Bal, and A. Destexhe, Comparison of different neuron models to conductance-based post-stimulus time histograms obtained in cortical pyramidal cells using dynamic-clamp *in vitro*, Biol. Cybern. **105**, 167 (2011).

[79] M. Monteforte, Chaotic dynamics in networks of spiking neurons in the balanced state, dissertation, University of Göttingen, 2011.

[80] R. Engelken, Chaotic neural circuit dynamics, dissertation, University of Göttingen, 2017.

[81] A. Viriyopase, R.-M. Memmesheimer, and S. Gielen, Analyzing the competition of gamma rhythms with delayed

pulse-coupled oscillators in phase representation, Phys. Rev. E **98**, 022217 (2018).

[82] P. Manz, S. Goedeke, and R.-M. Memmesheimer, Dynamics and computation in mixed networks containing neurons that accelerate towards spiking, Phys. Rev. E **100**, 042404 (2019).

[83] H. Sompolinsky, A. Crisanti, and H. J. Sommers, Chaos in random neural networks., Phys. Rev. Lett. **61**, 259 (1988).

[84] C. van Vreeswijk and H. Sompolinsky, Chaos in neuronal networks with balanced excitatory and inhibitory activity, Science **274**, 1724 (1996).

[85] S. Jahnke, R.-M. Memmesheimer, and M. Timme, How chaotic is the balanced state?, Front. Comput. Neurosci. **3**, 13 (2009).

[86] M. Monteforte and F. Wolf, Dynamical entropy production in spiking neuron networks in the balanced state, Phys. Rev. Lett. **105**, 268104 (2010).

[87] R. Engelken, F. Wolf, and L. F. Abbott, Lyapunov spectra of chaotic recurrent neural networks, Phys. Rev. Res. **5**, 043044 (2023).

[88] Z. Nadásdy, H. Hirase, A. Czurkó, J. Csicsvari, and G. Buzsáki, Replay and time compression of recurring spike sequences in the hippocampus, J. Neurosci. **19**, 9497 (1999).

[89] R. S. Johansson and I. Birznieks, First spikes in ensembles of human tactile afferents code complex spatial fingertip events, Nat. Neurosci. **7**, 170 (2004).

[90] A. Luczak, P. Barthó, and K. D. Harris, Spontaneous events outline the realm of possible sensory responses in neocortical populations, Neuron **62**, 413 (2009).

[91] M. N. Havenith, S. Yu, J. Biederlack, N.-H. Chen, W. Singer, and D. Nikolic, Synchrony makes neurons fire in sequence, and stimulus properties determine who is ahead, J. Neurosci. **31**, 8570 (2011).

[92] A. Stella, P. Bouss, G. Palm, and S. Grün, Comparing surrogates to evaluate precisely timed higher-order spike correlations, eNeuro **9** (2022).

[93] C. Howarth, P. Gleeson, and D. Attwell, Updated energy budgets for neural computation in the neocortex and cerebellum, J. Cereb. Blood Flow Metab. **32**, 1222 (2012).

[94] B. Olshausen and D. Fields, Sparse coding of sensory inputs, Curr. Opin. Neurobiol. **14**, 481 (2004).

[95] T. Flesch, K. Juechems, T. Dumbalska, A. Saxe, and C. Summerfield, Orthogonal representations for robust context-dependent task performance in brains and neural networks, Neuron **110**, 1258 (2022).