Journal of Signal Processing Systems (2022) 94:917–927
https://doi.org/10.1007/s11265-022-01772-5

Efficient Neuromorphic Signal Processing with Resonator Neurons

E. Paxon Frady[1] · Sophia Sanborn[1] · Sumit Bam Shrestha[1] · Daniel Ben Dayan Rubin[1] · Garrick Orchard[1] · Friedrich T. Sommer[1] · Mike Davies[1]

Received: 27 December 2021 / Revised: 11 April 2022 / Accepted: 2 May 2022 / Published online: 26 May 2022

Abstract
The biologically inspired spiking neurons used in neuromorphic computing are nonlinear filters with dynamic state variables, which is distinct from the stateless neuron models used in deep learning. The new version of Intel's neuromorphic research processor, Loihi 2, supports an extended range of stateful spiking neuron models with programmable dynamics. Here, we showcase advanced neuron models that can be used to efficiently process streaming data in simulation experiments on emulated Loihi 2 hardware. In one example, Resonate-and-Fire (RF) neurons are used to compute the Short Time Fourier Transform (STFT) with similar computational complexity but 47x less output bandwidth than the conventional STFT. In another example, we describe an algorithm for optical flow estimation using spatiotemporal RF neurons that requires over 90x fewer operations than a conventional DNN-based solution. We also demonstrate backpropagation methods to train non-linear spiking RF neurons for audio classification tasks, suitable for efficient execution on Loihi 2. We conclude with another application of nonlinear filtering showing a cascade of Hopf resonators exhibiting computational properties seen in the cochlea, such as self-normalization. Taken together, this work presents new techniques for an efficient spike-based spectrogram encoder that can be used for signal processing applications.

Keywords Neuromorphic computing · Resonator neurons · Spiking neural networks · Optic flow · Speech recognition

1 Introduction

In the language of signal processing, biological neurons are nonlinear time-varying filters. When interconnected in great numbers, biological neurons give rise to perception and intelligence, with remarkable energy efficiency. The recent successes of deep learning hints at the potential power of neural networks for signal processing, but the characteristics of artificial neural networks have diverged from their biological counterparts. Conventional artificial neuron models are vastly simplified compared to biology, with the rich temporal dynamics of biological neurons reduced to a point-wise nonlinearity, such as the ReLU function.

Neuromorphic chips, such as Intel's Loihi [1], can implement neuron models with a dynamical behavior similar to biological neurons, and their sparse communication and connectivity features support efficient scaling to high dimensional processing. In many cases, Loihi provides orders of magnitude gains in speed and energy compared to conventional solutions [2]. Today, the value of neuromorphic networks as signal processors remains underappreciated and underexplored.

With Loihi 2, we have augmented Loihi with enhancements aimed at expanding the breadth of signal processing problems that neuromorphic architectures can support. This paper discusses some of the enhancements and shares early examples that showcase the value of Loihi 2's richer feature set for intelligent and efficient signal processing applications. In particular, we highlight how resonating neuron models can be used for signal processing applications.

After introducing Loihi 2 in the next section, we demonstrate in simulation experiments different types of linear and nonlinear resonator units. First, we describe how resonate-and-fire (RF) neurons can be used in linear filtering applications of streaming signals. We show how these neurons can efficiently compute the Short-Time Fourier Transform (STFT) on audio signals, and how the coefficients can be encoded and processed with spikes. We then demonstrate

✉ E. Paxon Frady
e.paxon.frady@intel.com

Mike Davies
mike.davies@intel.com

[1] Intel Labs, Intel Corporation, Santa Clara, CA 95054, USA

how RF neurons can be used in vision to estimate optical flow on event-data through spatio-temporal filters that are computed in the membrane dynamics. We further describe how networks of nonlinear RF neurons can be trained with backpropagation and present early results on the NTIDIGITS and Google Speech Commands datasets. Finally, we describe a cochlea model based on a cascade of Hopf resonators performing nonlinear spectral decomposition on audio data. In particular, we find the critical density at which self-normalizing gain control emerges.

2 Loihi Architecture

Spiking neurons have been modelled as first order differential equations since Hodgkin and Huxley. Early Neuromorphic Engineers mimicked the dynamics of biological spiking neurons in silicon using analog electronic circuits [3]. More recently, there has been a shift towards digital implementations, both in software and in silicon, which naturally gives rise to a discrete time formulation of the internal neuron dynamics. Both Loihi and its successor, Loihi 2, use this digital approach (see [1] for details).

Loihi implements the discrete Leaky Integrate and Fire (LIF) neuron model

$$a_i[t] = \sum_j w_{ij} s_j[t-1] \tag{1}$$

$$u_i[t] = \lambda_u u_i[t-1] + a_i[t] \tag{2}$$

$$v_i[t] = \lambda_v v_i[t-1] + u_i[t] \tag{3}$$

where $a_i[t]$ is the accumulated synaptic activation for timestep t, u_i and v_i represent the i^{th} neuron's current and voltage respectively, and λ_u and λ_v are the current and voltage decay. Whenever $v_i[t]$ exceeds threshold, a spike is generated ($s_i[t] = 1$) and the voltage variable is reset to zero ($v_i[t] = 0$). All states and parameters use fixed precision.

Loihi 2 introduces a more flexible microcode programmable neural engine. Users can allocate variables and execute a wide range of instructions organized as short programs using an assembly language. These programs have access to neural state memory, the accumulated synaptic input a_i for the present timestep, random bits for stochastic models, and a timestep counter for time-gated computation. The instruction set supports conditional branching, bitwise logic, and fixed-point arithmetic backed by hardware multipliers.

Within a core, memory limits the number of different neurons which can be implemented. By using lower precision neuron models, more neurons can be implemented within the same memory footprint, up to a maximum of 8192 per core. More complicated neurons can be implemented as longer programs which access multiple memory addresses for neural state and synaptic input, and pass information to each other through the persistent thread state. The microcode engine has a flexible but finite set of operations and a maximum number of operations that can be executed each timestep, which may limit the complexity of neuron models.

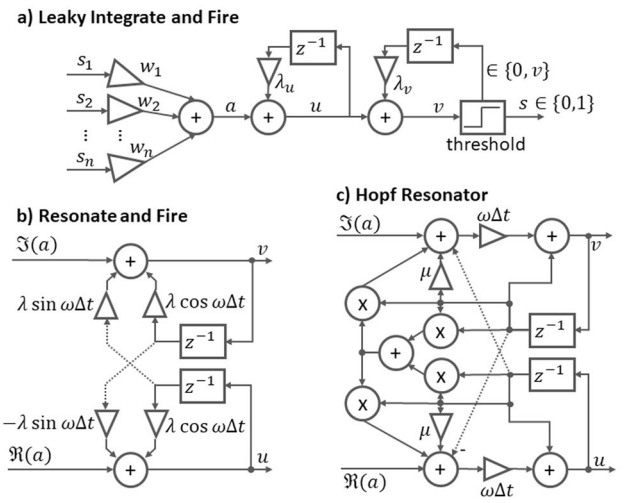

Figure 2 Block diagrams of the discrete computation for the LIF neuron, and RF, Hopf resonators. The Hopf Resonator shows an implementation of the Euler method, but in practice we use a 4th order Runge-Kutta.

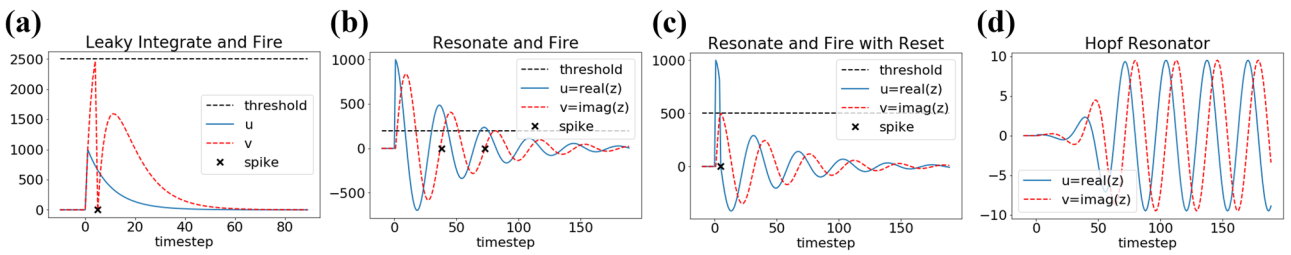

Figure 1 Response of neuron models used in this paper to an impulse (spike) at time 0. **a** A Leaky Integrate and Fire model which spikes whenever voltage exceeds threshold. **b** A complex valued Resonate and Fire model used for spatiotemporal filtering, which spikes whenever z crosses the real axis and $real(z)$ is greater than threshold. **c** A complex valued Resonate and Fire model with reset used in deep networks. **d** A complex valued Hopf Resonator with a stable limit cycle.

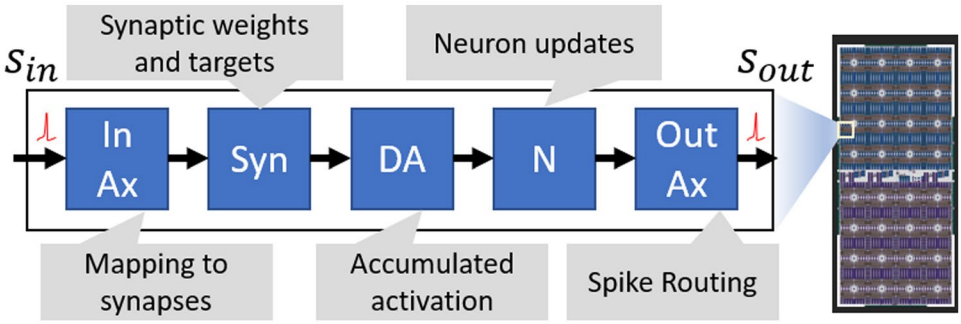

Figure 3 The Loihi 2 chip plot (right) and processing flow for a single core (left). Incoming spikes are mapped to lists of synapse weights which are accumulated for consumption in the next timestep. Meanwhile, neurons update using the previous timestep's accumulated activation and generate spikes which are routed to other cores by the Output Axon stage.

In addition to allowing much richer internal neuron dynamics, the neural engine allows user-defined output nonlinearities and reset mechanisms, and different from biological spiking, spikes in Loihi can carry a 32-bit integer payload, a format we refer to as *graded spikes*. The conditions determining when a neuron emits a spike, as well as the optional graded spike payload, are specified as part of the neuron's programming. Loihi 2 also supports *non-spiking* neurons that send a state variable, such as its membrane potential, unconditionally on every timestep or at regular sampling intervals. These are all new features in Loihi 2 which we make use of in this paper. Figure 1 shows the impulse (spike) response of the different Loihi 2 neuron models described in this paper, including their spiking output and reset behavior. Figure 2 shows the computation associated with these neuron models. While the LIF model uses a two stage cascade of filters, the RF and Hopf Resonator models use two cross-coupled filters to generate the real and imaginary components (see Fig. 3).

Loihi 2 further provides richer connectivity features than its predecessor. Synaptic activations can be computed from graded spikes, support for convolutional connections has been optimized, and new features allow procedural generation of stochastic synapses and separable synaptic matrices. Additional details of the Loihi 2 architecture can be found in the Loihi 2 Technology Brief[1].

3 Applications

3.1 Resonate-and-Fire Neurons for Spectral Analysis

The Resonate-and-Fire (RF) neuron is an extension of the standard LIF model, newly enabled in Loihi 2. The RF neuron is a damped harmonic oscillator with a spiking mechanism. The internal state is complex-valued $z = u + \mathrm{i}v$, where the variables u and v represent the real and imaginary parts. Each RF neuron is parameterized by a resonant frequency ω, and a decay factor $\lambda \in (0, 1)$, with dynamics defined by

$$z_k[t] = \lambda e^{\mathrm{i}\omega\Delta t} z_k[t-1] + a_k[t] \tag{4}$$

where the last term $a_k[t]$ is the synaptic input and $\lambda e^{\mathrm{i}\omega\Delta t}$ defines the oscillation kernel.

When starting from an initial condition of $z_k[t] = 0$ and assuming no reset mechanism, the dynamics of an RF neuron can be rewritten in the form

$$z_k[t] = \sum_n e^{\mathrm{i}n\omega\Delta t} \lambda^n a_k[t-n] \tag{5}$$

which is recognizable as one term for frequency ω of the discrete Short-Time Fourier Transform (STFT) of $a_k[t]$ with an exponential window. Thus, a bank of RF neurons each at different frequencies can then be used to compute the STFT.

The key feature of a spiking neuron is the temporally sparse pulsed output. Recently, it was shown how a spike-timing code can be used to represent the phases of complex variables [4]. Based on this, the complex-valued coefficients computed in the STFT can be encoded as a spike pattern. This is done by configuring a spiking condition for each RF neuron such that a spike is generated whenever the real-part exceeds a threshold as the imaginary-part crosses zero. By sending the magnitude $|z|$ as the payload of a graded spike, the combination of spike time and spike payload fully encodes the STFT complex coefficients. Conveniently, the magnitude is equivalent to the real-component of the state when the RF neuron's spike is triggered, which means no extra computation is required to transmit the magnitude. This encoding permits downstream neurons to potentially perform complex-valued matrix arithmetic.

Taken together, a bank of RF neurons naturally computes the STFT of an input signal through the internal dynamics and outputs the complex-valued coefficients as a timing pattern of graded spikes. By encoding a signal's spectrum in a sparse, event-driven manner with spikes, the communication bandwidth is automatically compressed without increasing latency. In the example shown in Fig. 4, the RF implementation reduces output bandwidth by 47x compared to a

[1] Taking Neuromorphic Computing to the Next Level with Loihi 2

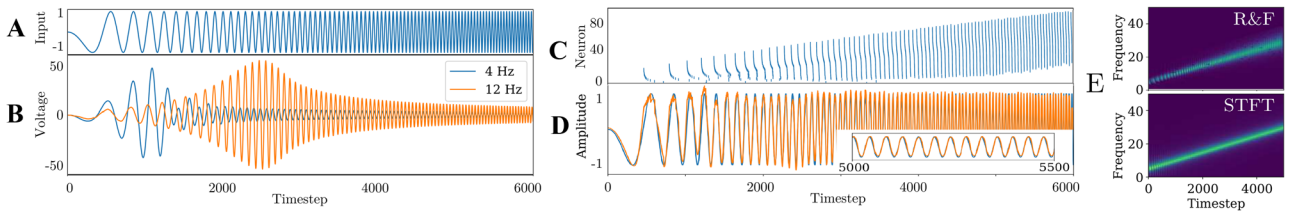

Figure 4 Using simulated RF neurons compatible with Loihi 2 to approximate the STFT of an audio chirp and reconstructing the input signal from the spike responses. **A**: The input chirp. **B**: Real component of the internal dynamics of two RF neurons with different resonant frequencies. **C**: Spike raster showing the output timing for all 100 neurons in response to the stimulus. **D**: Reconstruction (orange) of the original signal (blue) **E**: Comparison to the STFT.

conventional STFT producing a spectrogram vector on each time step. This advantage is partially because the RF neuron system naturally reduces output bandwidth proportional to the resonant frequency, akin to the wavelet transform [5].

We used RF neurons to produce STFTs for several examples in the Google Speech Commands dataset (Fig. 5). The Spiking STFT produced by the RF neurons can be inverted by convolving the graded spikes with the neuron's oscillation kernel and integrating across the population. We varied the spiking threshold of the RF neurons and measured the reconstruction correlation as a function of spikes produced. We compared this to the conventional STFT by excluding the smallest coefficients and performed inverse STFT to measure the reconstruction correlation. Reconstructions from the RF-generated spikes saturate to 94% correlation with only five thousand spikes. A conventional STFT computation generates over 3 million complex values over the same period. Computing a reconstruction from only the largest 500K nonzero coefficients maintains a high reconstruction correlation of 98%, but the reconstruction correlation drops to ~63% if only the largest 5,000 values are preserved. This shows that the RF implementation uses spike timing to encode the STFT information more efficiently than the conventional STFT.

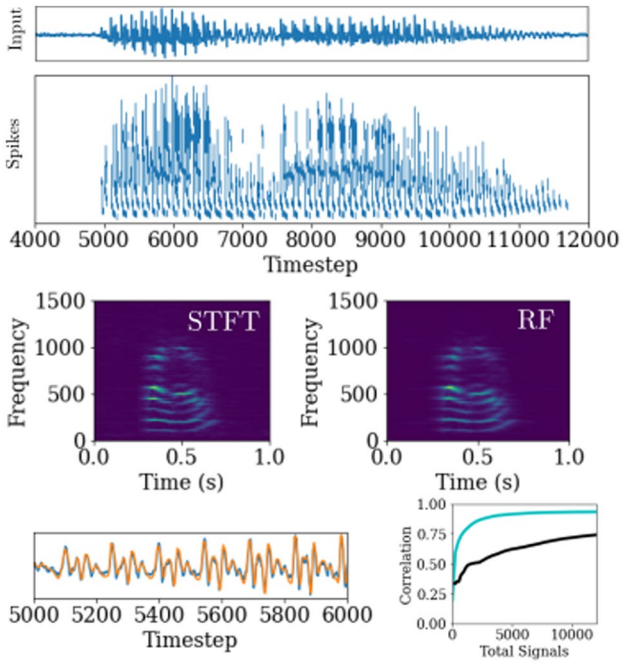

Figure 5 Spiking STFT for google speech command (top). Conventional STFT is shown compared to RF based STFT (middle). Reconstruction algorithm reproduces original signal (bottom). Reconstruction efficiency of RF model (cyan) is compared to thresholded conventional STFT (black).

3.2 Resonate-and-Fire Neurons for Estimating Optical Flow

While the membrane dynamics of a neuron can implement a temporal filter, the synapses to a neuron can implement a spatial filter. By combining the two we can compute separable spatiotemporal filters. Spatiotemporal filtering of this type has many potential signal processing applications, and it is especially efficient on neuromorphic hardware when processing sparse spiking data produced by other spiking neurons, or by Dynamic Vision Sensors (DVS) [6]. As a showcase, we demonstrate how RF neurons can be used to compute spatiotemporal filters on standard video data or streaming event data, and to estimate optical flow with motion energy algorithms [7].

The opponent energy model [7] consists of separate spatial and temporal filters, each with either even (blue) or odd (orange) symmetry (Fig. 6, left). This is mathematically equivalent to applying complex-valued filters to the input and the computation can be carried out with RF neurons (Fig. 6, right).

The spatiotemporal filters used in the model are complex-valued Gabor filters which have a preferred spatial frequency ω_x and orientation θ (Fig. 7). The temporal frequency of the filter ω_t is inherited from the resonant frequency of the RF neuron. The filter values are encoded as complex-valued synaptic weights that feed into the RF neurons.

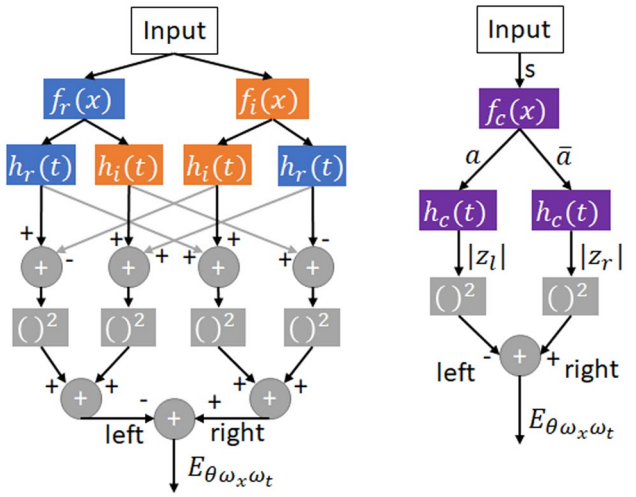

Figure 6 The original Adelson and Bergen Opponent Energy model (left) consisting of separable spatial $f()$ and temporal $h()$ components with orange filters out of phase with the blue filters. The equivalent model implemented using complex (purple) spatial and temporal filters.

Input pixel values or events from the DVS sensor, represented as s in Fig. 6, flow through the complex filters into a layer of resonate-and-fire neurons. For DVS input events, each input spike results in the complex-valued synaptic weight to be integrated by an RF neuron, where the real and imaginary parts of the weight are directly added to the real and imaginary state of the RF neuron. In the case of frames, a separate graded spike is used to represent the intensity of each pixel, and the activation a is the accumulated dot-product between the filter and input frame, which then gets integrated by an RF neuron. The filter with opposite preferred direction is computed by a second RF neuron that integrates the complex conjugate of the input activation $\bar{a}$, simplifying the computation and storage requirements of the filters.

The dynamics of the RF neuron integrating the complex-valued synaptic inputs then directly computes the coefficient of each spatiotemporal filter. If the input spikes are not

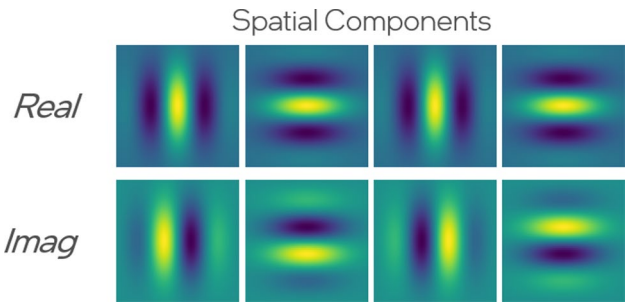

Figure 7 The complex-valued spatial filters combined with the RF neuron dynamics is used for spatiotemporal filtering.

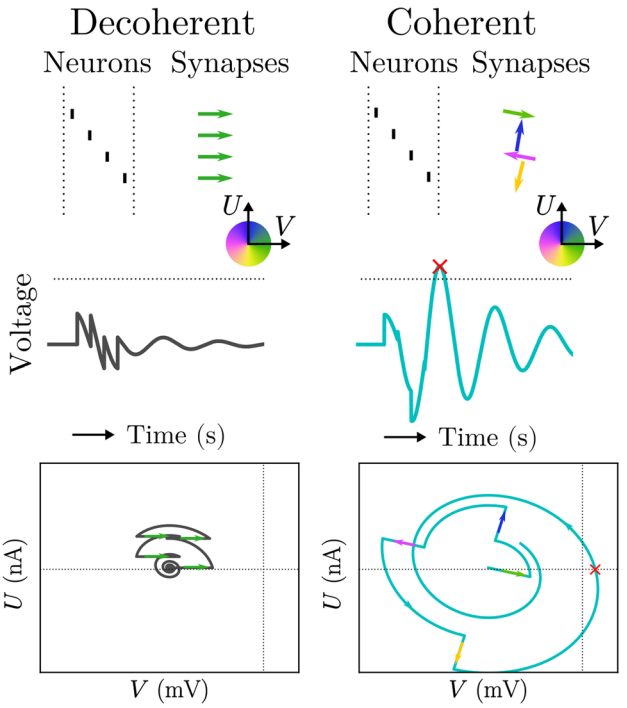

Figure 8 Integration of complex filters through dynamics of RF neurons. Input spikes (black) cause jumps in RF neuron's membrane potential based on synaptic weight (colored arrows).

aligned with the spatiotemporal pattern of the filter, then the RF neuron will have small sub-threshold dynamics (Fig. 8, left), and no spike indicates 0 as the filter coefficient. But if the input is aligned with the filter, then each input causes the magnitude of the internal state to grow. If sufficiently coherent, then the neuron will output a graded spike. The timing of the output spike signals the phase of the complex filter coefficient, and the graded value carries the magnitude.

The filters are applied across the image as a convolution, using Loihi 2's support for weight reuse of convolutional kernels. Input pixel events are filtered by shared kernel weights activating unique RF neurons at different temporal frequencies that are spatially mapped across the image. This provides a resource efficient implementation, which is important given Loihi 2's limited synaptic memory capacity.

After filtering, the motion energy model relies only on the magnitudes of the complex filter coefficients. First, the opponent energy $E_{\theta\omega_x\omega_t}$ is calculated based on the difference in magnitude of oppositely tuned filters. This difference can be computed by a downstream neuron that receives the graded input spikes of the two corresponding RF neurons. Loihi 2 supports a more efficient implementation in which a single neuron program directly implements the full opponent energy calculation. In this instantiation, a single neuron program maintains the state variables of two oppositely-tuned

Table 1 Optical flow model parameters.

Parameter	Units	Symbol	Count	Values
Receptive Field Size	pix	-	-	(64, 64)
Timestep Duration	sec	Δt	-	0.032
Spatial Frequency	rad/pix	ω_x	$n_x = 1$	$\omega_x = \frac{6\pi}{256}$
Temporal Frequency	rad/sec	ω_t	$n_t = 5$	$\omega_{t_k} = 4\pi k$
Orientations	rad	θ	$n_\theta = 4$	$\theta_k = \frac{k\pi}{n_\theta}$

RF neurons, and directly outputs the opponent energy values as a graded spike.

Once the opponent energy E is calculated for one combination of orientation θ, spatial frequency ω_x, and temporal frequency ω_t, the opponent energies from multiple filters of different frequencies and orientations must be combined to arrive at an optical flow estimate. We perform this step in post-processing off-chip. Each RF neuron with spatial frequency ω_x, temporal frequency ω_t and orientation θ has a preferred input velocity vector $v \in \mathbb{R}^2$, which is orthogonal to the orientation of the neuron's spatial receptive field:

$$v_{\omega_x,\omega_t,\theta} = \left[\frac{\omega_t}{\omega_x}\cos\theta,\ \frac{\omega_t}{\omega_x}\sin\theta\right] \tag{6}$$

Stimuli moving at the neuron's preferred velocity and direction produce the highest magnitude inner product with its spatiotemporal receptive field. To estimate the optical flow $f \in \mathbb{R}^2$ at a given pixel location, the neurons' preferred velocities are weighted by their normalized opponent energy at that pixel location:

$$f = \frac{\sum_{\omega_x,\omega_t,\theta} v_{\omega_x,\omega_t,\theta} E_{\omega_x,\omega_t,\theta}}{\sum_{\omega_x,\omega_t,\theta} E_{\omega_x,\omega_t,\theta}} \tag{7}$$

We implement spatiotemporal filters with Gabor shaped receptive fields of different orientations using the parameters in Table 1 to estimate optical flow from event data. The model was evaluated on the Multi View Stereo Event Camera (MVSEC) dataset [8] and compared to EV-FlowNet, a state-of-the-art model for estimating optical flow from event data. EV-FlowNet is a deep stateless neural network trained under self-supervision. Our model, in contrast, requires no training data, and processes the event data as it arrives, rather than buffering input data and processing it as voxels in a spatiotemporal tensor.

We compare to EV-FlowNet$_{2R}$, as it achieves the best performance across the test sequences (see [8] for details of EV-FlowNet). We evaluate the models using the methods described in [8], calculating the Average Endpoint Error (AEE; the average distance between computed flow vectors f and the ground-truth flow vectors) and outlier percentage (percentage of flow vectors with AEE > 3 pixels).

Table 2 compares results for two versions of our model. The *dense* version estimates flow directly from the neuron's internal state in a single Loihi 2 program. The *spikes* version uses only the graded spikes that are output from RF neurons. Across the three indoor flying sequences from [8], our models achieves better performance than EV-FlowNet on these two metrics. Figure 9 illustrates a single representative frame.

Compared to EV-FlowNet, our model has less than half the neurons, although each RF neuron update is more expensive (4 MACs versus 1 ReLU). However, computation in EV-FlowNet is dominated by synaptic ops, which outnumber the neuron updates by 2000x. Our model has over 10x fewer synapses, our synaptic ops are cheaper (a complex AC versus a MAC), and sparse synapse activation results in 93x fewer synaptic ops on the MVSEC sequences tested.

To compute optical flow from the neuron state, the neuron must square its magnitude to compute the opponent energy (Fig. 6), and the energies must be combined in post-processing off-chip. Off-chip post-processing (6) introduces another 2 MACs per neuron and one inversion per pixel, which is still dwarfed by the synaptic ops. For comparison, readout neurons in EV-FlowNet require computing tanh twice per pixel.

The RF optical flow model benefits from two key properties. The event-based operation exploits sparsity of the input spike data to reduce synaptic ops, and an overlapping region between two subsequent temporal windows only needs to be processed once, saving further ops.

3.3 Using Backpropagation to Train RF Neurons

We have extended the Spike Layer Error Reassignment (SLAYER) tool [9], which was previously used to train LIF neural networks on Loihi, to handle complex and oscillatory models with graded spikes, including the RF neuron, for Loihi 2. This new implementation of SLAYER is available

Table 2 Average Endpoint Error on MVSEC.

	Indoor Flying 1		Indoor Flying 2		Indoor Flying 3	
	AEE	% outlier	AEE	% outlier	AEE	% outlier
EV − FlowNet$_{2R}$	1.03	2.2	1.72	15.1	1.53	11.9
Ours$_{\text{DENSE}}$	0.91	**0.35**	1.28	5.83	1.04	2.88
Ours$_{\text{SPIKES}}$	**0.83**	0.68	**1.22**	**5.42**	**0.97**	**2.65**

Bold indicates best performing model

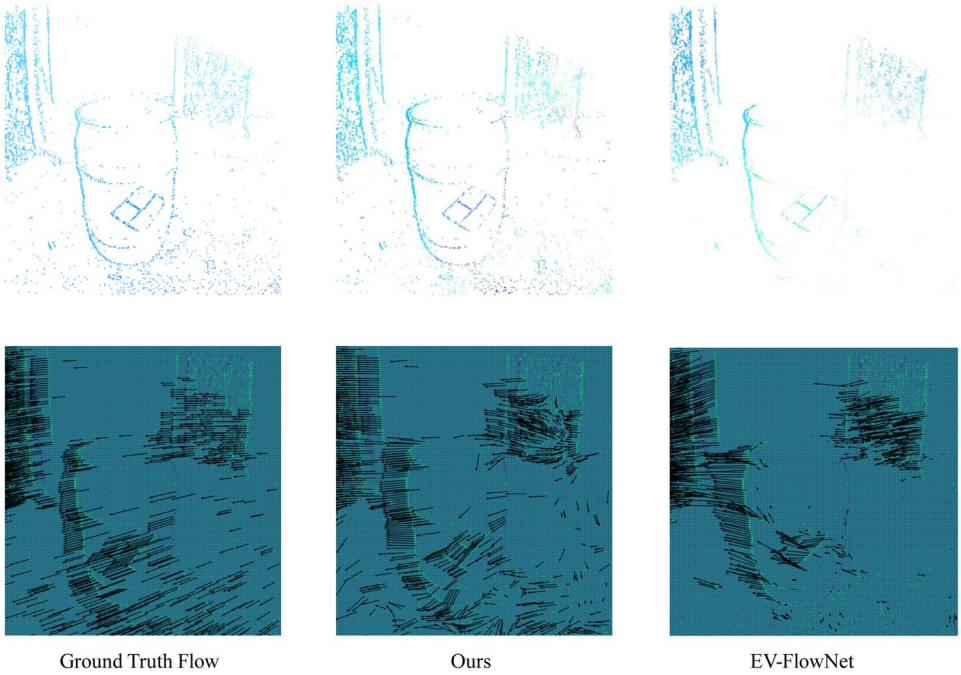

Figure 9 Ground truth optical flow from the MVSEC dataset (left) compared to our method (middle) and EV-FlowNet (right).

publicly as part of the Deep Learning library in Lava, an open source neuromorphic software framework[2].

The extension generalizes the temporal error credit assignment problem for neuron dynamics formulated as a linear time-invariant system. For the specific case of RF neuron dynamics represented by the decaying rotation operator $\lambda e^{-i\omega\Delta t}$, the temporal error credit assignment is simply the decaying rotation operator applied to gradients backward in time. This is illustrated in Fig. 10. What is different in temporal credit assignment for RF dynamics compared LIF neurons is that the error credit sign switches at different points in history according to the phase of the RF rotation dynamics.

A surrogate function is typically used to approximate the function derivative for gradient backpropagation in spiking neurons [9, 10]. We extend the concept to relaxation of the actual Dirac-Delta function in the derivative term using nascent delta approximation.

$$\frac{\partial \mathcal{H}(x)}{\partial x} = \delta(x) \approx \phi_n(x) \tag{8}$$

where $\phi_n(\cdot)$ is the nascent delta approximation. One example is $\phi_n(x) = \frac{1}{a_n} e^{-\frac{|x|}{a_n}}$. This generalization applies to the phase spiking mechanism described in Sect. 3.1.

The RF neurons described in Sect. 3.1, are essentially linear filters. For deep networks, we introduce an output nonlinearity for the RF neuron following the model proposed in [11]. In this model, the RF neuron generates a spike whenever its imaginary component exceeds the threshold, following which the real component is reset to 0 (Fig. 1c). We find that this nonlinear spiking mechanism has better learning behavior with backpropagation.

Using SLAYER we trained a hybrid MLP of RF and LIF neurons (64-256RF-256RF-242LIF) on the spiking NTIDIGITS [12] audio dataset. The model, with 226K parameters, predicts digit utterances with an accuracy of $92.14 \pm 0.24\%$. In contrast, the best-known conventional solution using LSTM units (643K parameters) achieves an accuracy of 91.25% [12].

We also tackled the more challenging Google Speech Commands dataset [13] (10+2 subset) with both MLP and convolutional architectures by first converting the dataset to

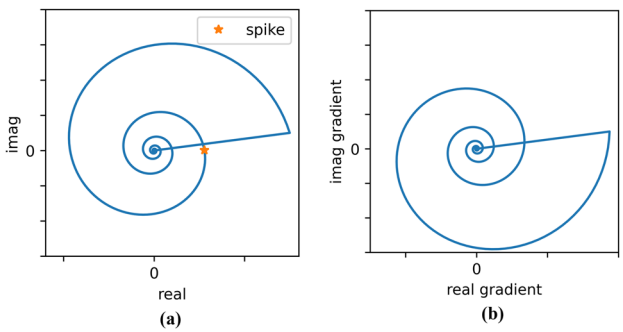

Figure 10 Complex phase plot showing **a** forward dynamics of resonate and fire neuron. **b** temporal credit assignment of gradients accounting for resonate and fire neuron dynamics.

[2] https://lava-nc.org. The `lava-dl` deep learning library is available at https://github.com/lava-nc/lava-dl

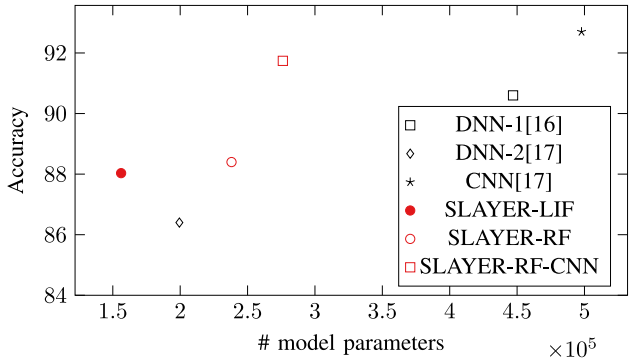

Figure 11 Performance comparison of feed-forward networks on Google Speech Commands 10+2 subset task [16, 17].

spikes using a publicly available cochlea model [14]. The MLP architecture (64-256RF-256RF-288LIF) has 238K parameters and achieves an accuracy of 88.97%. An equivalent LIF only architecture with 156K parameters achieved an accuracy of 88.03%. A hybrid CNN architecture with five RF convolution layers followed by two LIF dense layers, achieves 91.74%.

The performance of SLAYER trained networks with other existing feed-forward streaming models on Google Speech Commands is shown in Fig. 11. It is worth noting that the hybrid CNN architecture shows competitive performance with sparse spike based communication even compared to best performing CNN while requiring about half the parameters. The hybrid CNN's accuracy is comparable to the best known recurrent SNN result of 92.10% [15] with 285K parameters even though it is a feed-forward network. We acknowledge that non-event based recurrent networks and attention based networks yield better accuracy on Google Speech Commands. One such example is the Legendre Memory Unit which shows an accuracy of 96.9% while requiring just 97K parameters (recurrent networks have an advantage in parameter count, since they can reuse the same weights in the recurrent computations). We expect to close the performance gap with further improvements and the ability to train recurrent RF networks in the future.

3.4 Cascaded Hopf Resonators

Loihi 2's programmable architecture allows neuron models to unconditionally generate graded spikes at regular intervals, which may be interpreted as samples of analog waveforms. An additional nonlinear feedback term can also be added to the RF neuron dynamics yielding the Hopf resonator (Fig. 1(c)). In particular, the feedback consists of the squared modulus of the complex state variable (9) resulting in a cubic nonlinear term that persists even for very weak stimuli referred to as "essential nonlinearity" [18]. As the stimuli become stronger, the cubic term increases at about the same rate so that the nonlinear harmonics produced remain almost constant. This behavior is very different from a passive nonlinear system and corresponds to an oscillator operating near a Hopf bifurcation. A Hopf bifurcation is a critical point at which a periodic solution to the differential equation arises because the system stability switches (Fig. 1(d)). The distance to the critical point is adjusted by input strength a which offers self-adjusting gain and bandwidth control.

Hopf dynamics have wide-ranging signal processing applications, a classical one being regenerative receivers, which have recently seen a revival in low-power wireless applications [19]. Hopf resonators are also used as active elements in models of the auditory pathway up to the auditory nerve [20, 21], providing biologically inspired methods for audio preprocessing. As signal processing units on Loihi 2, these approaches promise simplicity, compactness, and efficiency thanks to their many emergent properties.

One model of the cochlea we consider here for Loihi 2 implementation uses a cascade of Hopf resonators to represent basilar membrane sections at different frequencies [20]. Specifically, each membrane section provides a band-passed filtered version of its input to the following lower frequency section, and so on.

We begin by modeling a cochlea section using the continuous formulation

$$\dot{z} = \omega_0((\lambda - |z|^2 + i)z + a), \tag{9}$$

where $z \in \mathbb{C}$ is the resonator response, $\omega_0 \in \mathbb{R}$ is the characteristic frequency of the cochlea section, $\lambda \in \mathbb{R}$ is a parameter that controls the Hopf bifurcation, and $a \in \mathbb{C}$ is the external input to the resonator. The output of the model is a complex analog quantity. For a Loihi 2 implementation, we consider a discrete time version of (9) shown in Fig. 1(c), discretized using a 4th order Runge-Kutta method.

We have explored different densities of sections per octave and have found that a self-normalizing gain control emerges at a critical density. We here show two example densities $\delta = 2, 6$ the first one before and the second one after the critical density that occurs at $\delta \approx 5$. In order to show the emergent self-normalization effect, we first need to compute a selectivity plot for each resonator by sweeping a pure tone into the the highest frequency Hopf resonator unit at the beginning of the cochlea cascade. The output of the highest frequency resonator is then fed to the next highest resonator, and so on. By recording the peak of the temporal response to each resonator's incoming signal, we obtain Fig. 12(a, b). Each subplot is showing the selectivity of each resonator in cascade to an input tone with amplitude $A = \{0.01, 0.08, 2, 10\}$. If we now take the envelope of the peak responses of the selectivity plot at constant amplitude and density, then we obtain a maximum

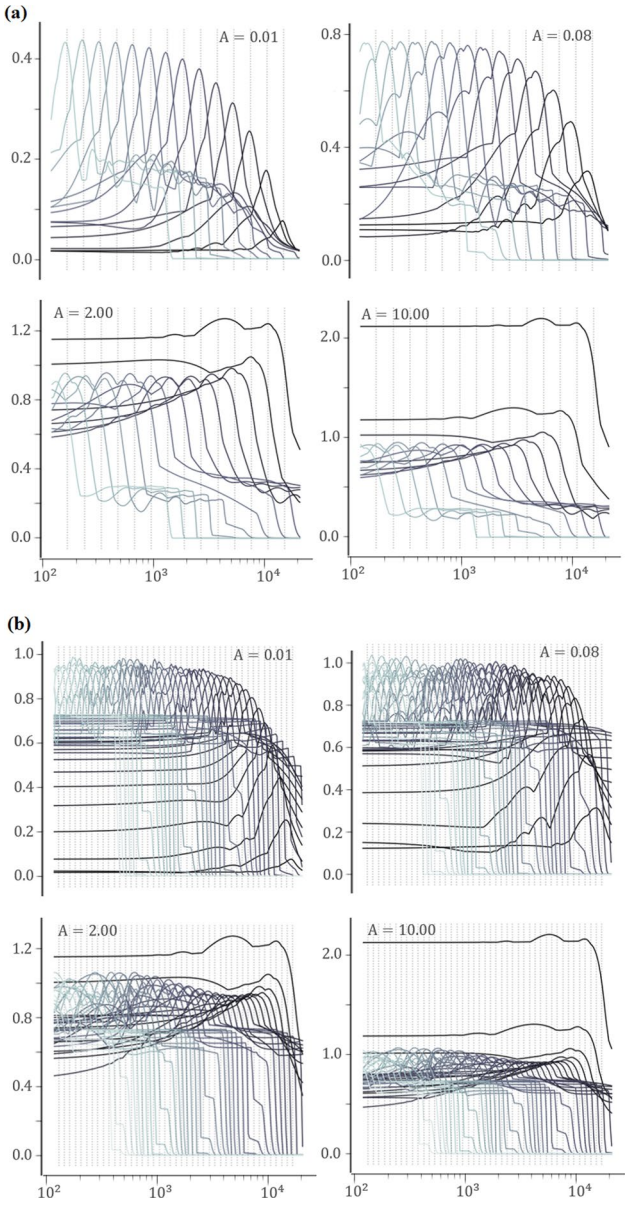

Figure 12 Cascade of Hopf resonators for various amplitudes (0.01, 0.08, 2, 10) showing the selectivity of each resonator to incoming tones (from dark to lighter solid lines for resonator with lower characteristic frequency). **a** density of 2 sections per octave **b** 6 sections/octave density.

peak contour for each input tone amplitude along the cascade (Fig. 13). At higher densities, the cascading provides a *self-normalizing gain* control – signal amplitudes across several orders of magnitude of frequency range are normalized to a narrow dynamic range (-3dB, 0dB). This self-normalizing gain control is a novel observation, an emergent property of cascading Hopf resonators at densities larger than 4 per octave. For implementation on Loihi 2 and other hardware platforms, this represents a very manageable density requirement, e.g. a total of no more than 30 resonators is needed to cover the entire human auditory range of 5-6 octaves.

4 Discussion

Loihi 2's generalized feature set, including a programmable neuron engine, provides far greater flexibility than Loihi for exploring novel spiking neural network models. In this paper, we have described oscillatory neuron models which can be executed on Loihi 2 that go beyond the much-studied leaky-integrate-and-fire model. The simulation experiments show several potential signal processing applications that can be computed using oscillatory neuron models. Using these techniques, efficient signal processing can be performed on neuromorphic hardware, providing advantages for latency and energy constrained applications.

The STFT is widely used in signal processing applications for non-stationary signals. The real-time and high throughput requirements of these applications make parallel implementations essential. In terms of computational complexity, the graded spike RF network shares the efficiency of previous proposals of parallel hardware implementations of the STFT based on iterative integration of inputs [22]. By encoding the complex coefficients with a graded spike-timing code, we also see great reductions in the bandwidth needed to encode the signal's spectral representation and expect similar reductions in downstream processing on Loihi 2.

The optical flow algorithm we presented is a reformulation of the Adelson & Bergen model [7]. Generally, spatiotemporal filters can be designed in the complex domain, and we described how the oscillatory dynamics of RF neurons

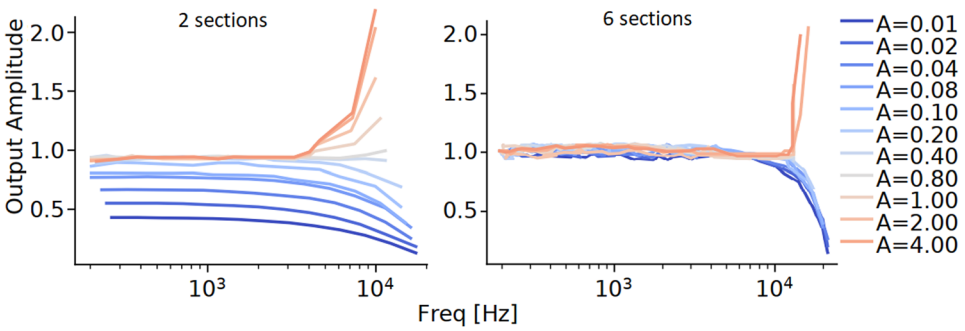

Figure 13 Amplitude normalization using a cascade of Hopf resonators. Peak output amplitude (y-axis) for a cascade of Hopf resonators plotted against input stimulus frequency (x-axis) for different input amplitudes (**A**). Increasing the number of sections per octave from 2 (left) to 6 (right) increases the sharpness of the normalization tuning.

can compute these filters on streaming video or the sparse events produced by neuromorphic cameras. In conjunction with efficient convolutions available on Loihi 2, this approach is a simple and efficient way to process video data with spatiotemporal filters. We compared this model to a multi-layer convolutional network for estimating optical flow. This convolutional network was not designed with any efficiency or simplicity goals, yet the very simple model presented here reached similar or better performance levels.

The dynamics of the RF neurons, along with the encoding of complex coefficients as a graded timing pattern of spikes, leads to a system which performs linear complex-valued matrix operations. The STFT and optical flow applications are essentially linear filtering operations, with only a slight nonlinearity in which amplitudes below threshold are not transmitted.

More generally, RF neurons can be modified to perform nonlinear transformations. In both deep learning applications and auditory processing applications inspired by the cochlea, nonlinearities are essential. The nonlinearity can be introduced by including higher-order terms in the membrane dynamics, or by the spiking and reset mechanisms. Other nonlinearities can be used for signal processing algorithms, for example, binary spikes [4], or thresholded graded spikes, can be used for nonlinear compression, i.e., clipping, of STFT representations.

We have demonstrated that networks of nonlinear RF neuron models can be trained using backpropagation in the deep learning paradigm, similar to recent work on the training of networks of LIF neurons [9]. We compared our results to other approaches that classify streaming data, and achieve similar performance. While these feed-forward RF networks do not yet achieve state-of-the-art performance, these are nascent models and improvements may be expected as training methods mature, for example with support for recurrent architectures.

Another example of nonlinear signal processing supported by Loihi 2 is the cascaded-Hopf resonator, in which oscillatory units with different resonant frequencies are coupled, and a higher-order feedback term is included in their dynamics. In our model inspired by the basilar membrane, we have shown that a sufficiently high density of Hopf resonators per octave leads to peak self-normalization, a novel emergent property on top of the nonlinear spectral decomposition of the model. The Hopf resonator cascade offers a compact self-normalized spectral decomposition that could become an alternative for STFT or MFCC preprocessing in future audio applications.

Declarations

Competing Interest The authors are employees of Intel Labs. Sophia Sanborn contributed to this research while at Intel, and is currently at UC Berkeley. On behalf of Intel, Daniel Ben Dayan Rubin filed a patent application 17/643,652, AD7779-US named "Peak self-normalization gain control based on Hopf resonators cascade signal spectral decomposition". The authors declare no additional competing interests. E. Paxon Frady, Sophia Sanborn, Sumit Bam Shrestha, Daniel Ben Dayan Rubin, performed experiments and analysis. E. Paxon Frady, Sumit Bam Shreshtha, Daniel Ben Dayan Rubin, Garrick Orchard, Friedrich T. Sommer, and Mike Davies wrote the paper.

References

1. Davies, M., Srinivasa, N., Lin, T., Chinya, G., Cao, Y., Choday, S. H., Dimou, G., Joshi, P., Imam, N., Jain, S., Liao, Y., Lin, C., Lines, A., Liu, R., Mathaikutty, D., McCoy, S., Paul, A., Tse, J., Venkataramanan, G., Weng, Y., Wild, A., Yang, Y., & Wang, H. (2018). Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro, 38*(1), 82–99.
2. Davies, M., Wild, A., Orchard, G., Sandamirskaya, Y., Guerra, G. A. F., Joshi, P., Plank, P., & Risbud, S. R. (2021). Advancing neuromorphic computing with Loihi: A survey of results and outlook. *Proceedings of the IEEE, 109*(5), 911–934.
3. Arthur, J. V., & Boahen, K. A. (2010). Silicon-neuron design: A dynamical systems approach. *IEEE Transactions on Circuits and Systems I, 58*(5), 1034–1043.
4. Frady, E. P., & Sommer, F. T. (2019). Robust computation with rhythmic spike patterns. *PNAS, 116*(36), 18050–18059.
5. Polikar, R. (1996). Fundamental concepts & an overview of the wavelet theory.
6. Gallego, G., Delbruck, T., Orchard, G. M., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A., Conradt, J., Daniilidis, K., & Scaramuzza, D. (2020). *Event-based vision: A survey*. IEEE Transactions on Pattern Analysis and Machine Intelligence.
7. Adelson, E. H., & Bergen, J. R. (1985). Spatiotemporal energy models for the perception of motion. *Journal of Optical Society of America, 2*(2), 284–299.
8. Zihao Zhu, A., Thakur, D., Özaslan, T., Pfrommer, B., Kumar, V., & Daniilidis, K. (2018). The multivehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robotics and Automation Letters, 3*(3), 2032–2039.
9. Shrestha, S. B., & Orchard, G. (2018). SLAYER: Spike layer error reassignment in time. In *Advances in Neural Information Processing Systems (NeurIPS)* (pp. 1412–1421).
10. Neftci, E. O., Mostafa, H., & Zenke, F. (2019). Surrogate gradient learning in spiking neural networks. *IEEE Signal Processing Magazine, 36,* 61–63.
11. Izhikevich, E. M. (2001). Resonate-and-fire neurons. *Neural Networks, 14*(6), 883–894.
12. Anumula, J., Neil, D., Delbruck, T., & Liu, S.-C. (2018). Feature representations for neuromorphic audio spike streams. *Frontiers in Neuroscience, 12*.
13. Warden, P. (2018). Speech commands: A dataset for limited-vocabulary speech recognition. arXiv preprint arXiv:1804.03209
14. Zilany, M. S., Bruce, I. C., & Carney, L. H. (2014). Updated parameters and expanded simulation options for a model of the auditory periphery. *The Journal of the Acoustical Society of America, 135*(1), 283–286.
15. Yin, B., Corradi, F., & Bohté, S. M. (2021). Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. preprint arXiv:2103.12593
16. Rybakov, O., Kononenko, N., Subrahmanya, N., Visontai, M., & Laurenzo, S. (2020). Streaming keyword spotting on mobile devices. In *Interspeech*.
17. Zhang, Y., Suda, N., Lai, L., & Chandra, V. (2017). Hello edge: Keyword spotting on microcontrollers. arXiv preprint arXiv:1711.07128

18. Goldstein, J. (1967). Auditory nonlinearity. *Journal of the Acoustical Society of America, 41,* 676–689.
19. Tapson, J., Hamilton, T. J., Jin, C., & van Schaik (2008). A. Self-tuned regenerative amplification and the HOPF bifurcation. In *IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1768–1771).
20. Eguíluz, V. M., Ospeck, M., Choe, Y., Hudspeth, A., & Magnasco, M. O. (2000). Essential nonlinearities in hearing. *Physical Review Letters, 84*(22), 5232–5235.
21. Kern, A., & Stoop, R. (2003). Essential role of couplings between hearing nonlinearities. *Physical Review Letters, 91*(12), 128101.
22. Liu, K. J. R. (1993). Novel Parallel architectures for short-time fourier transform. *IEEE TCS II, 40*(12), 786–790.