Leveraging sparsity with Spiking Recurrent Neural Networks for energy-efficient keyword spotting

Manon Dampfhoffer, Thomas Mesquida, Emmanuel Hardy, Alexandre Valentian, Lorena Anghel

LEVERAGING SPARSITY WITH SPIKING RECURRENT NEURAL NETWORKS FOR ENERGY-EFFICIENT KEYWORD SPOTTING

Manon Dampfhoffer[1,2], Thomas Mesquida[2], Emmanuel Hardy[3], Alexandre Valentian[2], Lorena Anghel[1]

[1]Univ. Grenoble Alpes, CEA, CNRS, Grenoble INP, INAC-Spintec, 38000 Grenoble, France
[2]Univ. Grenoble Alpes, CEA, List, F-38000 Grenoble, France
[3]Univ. Grenoble Alpes, CEA, Leti, F-38000 Grenoble, France

ABSTRACT

Bio-inspired Spiking Neural Networks (SNNs) are promising candidates to replace standard Artificial Neural Networks (ANNs) for energy-efficient keyword spotting (KWS) systems. In this work, we compare the trade-off between accuracy and energy-efficiency of a gated recurrent SNN (SpikGRU) with a standard Gated Recurrent Unit (GRU) on the Google Speech Command Dataset (GSCD) v2. We show that, by taking advantage of the sparse spiking activity of the SNN, both accuracy and energy-efficiency can be increased. Leveraging data sparsity by using spiking inputs, such as those produced by spiking audio feature extractors or dynamic sensors, can further improve energy-efficiency. We demonstrate state-of-the-art results for SNNs on GSCD v2 with up to 95.9% accuracy. Moreover, SpikGRU can achieve similar accuracy than GRU while reducing the number of operations by up to 82%.

Index Terms— Spiking neural networks, keyword spotting, speech commands, energy-efficiency, sparsity.

1. INTRODUCTION

Keyword spotting (KWS), which consists in detecting specific keywords in an audio stream comprising speech, has a wide range of applications such as activation of voice assistants, voice control, speech data mining, routing phone calls, etc. [1]. Artificial Neural Networks (ANNs) have shown impressive performance on these tasks, but their energy consumption limits their use in embedded systems. Indeed, always-on KWS systems for small electronic devices, such as activation of voice assistants, have power and energy constraints. Reducing the model size (number of neurons and parameters) is one solution to reduce the computational and memory load of ANNs [2, 3].

Bio-inspired Spiking Neural Networks (SNNs) are another promising research direction targeting the reduction of energy consumption in embedded hardware. Indeed, their computations are based on the accumulation of binary spikes instead of real-valued activations, which allows multiply-and-accumulate (MAC) operations to be replaced by accumulate (AC) operations consuming less energy [4]. Moreover, spikes can be handled in an event-based manner in neuromorphic hardware, allowing to exploit their sparsity [5].

Spiking fully connected or convolutional neural networks (CNNs) have been demonstrated for KWS using the Google Speech Command Dataset (GSCD) [6] v1 [7, 8, 9] and v2 [10]. However, spiking Recurrent Neural Networks (RNNs) have not yet been proposed for KWS, although RNNs are well suited for spatiotemporal data such as speech. Moreover, RNNs, such as Gated Recurrent Units (GRUs) [11], have shown high performance on low-power embedded hardware for KWS [12]. In addition, spiking recurrent topologies demonstrated higher energy and time savings on the Loihi neuromorphic chip than convolutional topologies [5].

Recently, we have proposed a recurrent SNN model, SpikGRU (Spiking Gated Recurrent Unit) [13] achieving high accuracy on spoken digits and words recognition tasks with high spike sparsity. Moreover, we have shown previously that sparsity is the main advantage of SNNs to increase their energy-efficiency compared to ANNs [14]. Motivated by these results, in this work we evaluate SpikGRU on the KWS task of GSCD v2 [6] and we investigate its accuracy-efficiency trade-off compared to GRU with different network sizes. We show the benefits of exploiting the sparsity in SpikGRU by regularizing the spiking activity during the training. Then, we explore the advantage of leveraging sparsity in the input data by converting the real-valued inputs into spikes. We achieve state-of-the-art results compared to previous works with SNNs. Moreover, SpikGRU with activity regularization shows significant improvement in energy-efficiency compared to GRU with the same accuracy.

2. METHODS

2.1. ANN and SNN models

We used SpikGRU [13] as our SNN baseline and GRU [11] as ANN baseline. SpikGRU uses a single gate z to control the information flow in the membrane potential v by selecting

Thanks to MIAI @ Grenoble Alpes, (ANR-19-P3IA-0003) for funding.

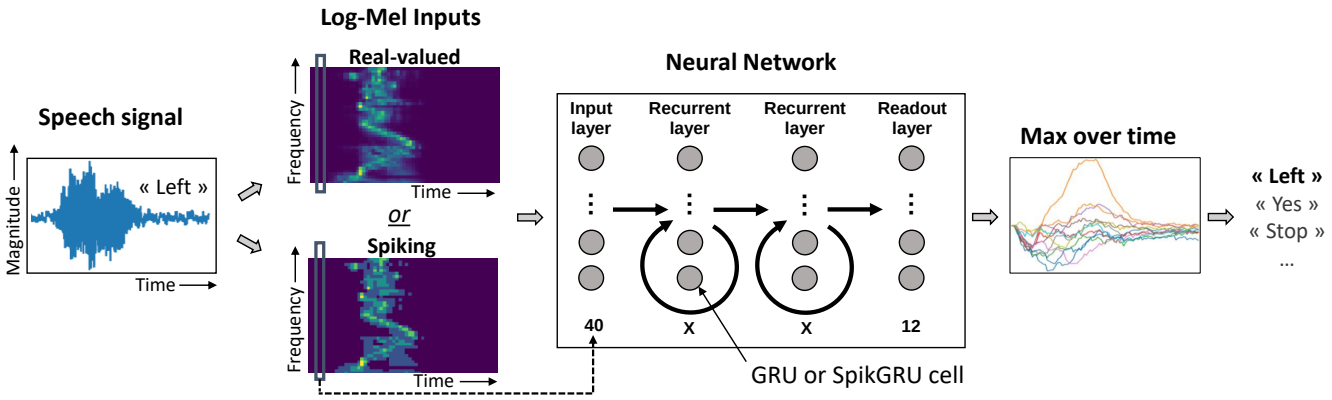

Fig. 1. Keyword spotting task. Log-Mel features are extracted from the raw audio signal and either the real values (experiment 1) or the converted spiking inputs (experiment 2) are fed to the neural network at each timestep. The neural network has two recurrent layers of X GRU or SpikGRU cells. The maximum value over time of the readout neurons is used for the prediction.

the best combination of its previous state and the input i. The neuron fires a spike when v reaches the threshold v_{th}. It is defined as follows [13]:

$$i_t^l = \alpha \odot i_{t-1}^l + W_i s_t^{l-1} + U_i s_{t-1}^l + b_i \quad (1)$$

$$z_t^l = \sigma(W_z s_t^{l-1} + U_z s_{t-1}^l + b_z) \quad (2)$$

$$v_t^l = z_t^l \odot v_{t-1}^l + (1 - z_t^l) \odot i_t^l - v_{th} s_{t-1}^l \quad (3)$$

$$s_t^l = H[v_t^l - v_{th}] \quad (4)$$

$\odot$ denotes element-wise multiplication, s_t^l are output spikes of neurons from layer l at time t, σ is sigmoid function and H is the Heaviside step function corresponding to spike firing. W_i, W_z and U_i, U_z are the weight matrices of feed-forward and recurrent connections respectively, b_i and b_z the biases and α a vector of time constants. Only binary spikes are communicated in the feedforward and recurrent connections. Thus, matrix multiplications are replaced by event-based accumulations, allowing to reduce the number of operations depending on the spike sparsity. However some element-wise multiplications are necessary as the internal variables of the model (i, z, v) are real-valued.

Although MAC and AC do not have the same energy consumption in hardware [4], we have used the total number of operations (MACs + ACs) as a hardware-independent figure of merit for energy-efficiency. The number of MACs and ACs per layer are computed using the input and output size of the layer, and the input and output spike activity rate (number of spikes per timestep per neuron) for SNNs as in [13].

2.2. Dataset and pre-processing

GSCD v2 contains 35 different words of at most 1 second sampled at 16 kHz. The keyword spotting task consists in a 12-class classification problem with 10 keywords (“yes”, “no”, “up”, “down”, “left”, “right”, “on”, “off”, “stop”, “go”), “silence” (background noise) and “unknown” (non-keyword words) classes. The dataset is provided with 84,843 training, 9,981 validation and 4,890 test samples.The pipeline used in these experiments is shown in Fig 1. From the audio signals we extracted 40 log-Mel features, with frequencies between 80Hz and 8kHz, window size of 30 ms and hop length of 10 ms. This results in 100 simulation timesteps with the values of the 40 channels being fed to SNNs and ANNs at each timestep. The input samples are re-scaled such that each channel has a unit variance across the time dimension. We used data augmentation with background noise and time shift as in [2], and time and frequency masking [15].

In the first experiment, we directly fed the real-valued (32b float) log-Mel to neural networks, while in the second experiment, we convert them to spikes. For the spike conversion, we scale each real value by a factor $\in \{0.5, 1.0, 1.5, 2.0\}$ and round the result to the closest integer. Thus, each channel can produce between 0 and $\in \{5, 11, 16, 21\}$ spikes per timestep (respectively), leading to an average (measured on the test set) input activity rate $\in \{0.23, 0.48, 0.74, 0.99\}$ spikes per channel per timestep (respectively).

2.3. Training procedure

ANNs and SNNs are composed of two recurrent layers with $X \in \{32, 64, 128, 256, 512\}$ units each, and a readout layer (fully-connected to the last recurrent layer) which is composed of leaky integrators in the case of SNNs. Indeed, we empirically found that, for the same number of parameters and activity, two layers yield better results than one but the improvement was less important for three layers. We used a max-over-time loss, which is the cross-entropy loss applied on the maximum value of the neurons of the readout layer over all timesteps. All models are trained with backpropagation through time for 100 epochs and a batch size of 128 using Adam optimizer with a learning rate starting from 0.001

and decaying to 0 with a cosine annealing scheduler. For the SNNs, weights and biases are initialized from a uniform distribution $U(-k^{-1/2}, k^{-1/2})$, k being the input size of the layer. The time constant α is a learnable parameter per neuron and initialized at 0.8 and v_{th} is set to 1. As the spiking activation function is not differentiable, we define a surrogate gradient using a piece-wise linear triangular function [16]. In all our experiments, we ran each configuration 5 times and reported the mean accuracy with the 95% confidence interval.

We investigated the effect of regularizing the spiking activity during the training of SNNs to decrease the number of spikes and thus the number of operations. We added a term to the loss function to penalize spike firing per layer as in [10]:

$$loss^{l}_{reg} = \frac{1}{2}\frac{1}{N}\frac{1}{T}\sum_{t}\sum_{n} S_n[t]^2 \quad (5)$$

with N the number of neurons in layer l, T the number of timesteps, and $S_n[t]$ equals to 1 if neuron n fired a spike at time t and 0 otherwise. This term is weighted by a coefficient $\lambda \in \{0.5, 1, 2, 4, 10, 50\}$ allowing us to adjust the impact of the penalization and thus the spiking activity in the network. In the second experiment, we use the λ leading to the maximal accuracy for each topology in the first experiment.

3. RESULTS

3.1. Results with Real-valued Inputs

We first evaluated SpikGRU and GRU on GSCD v2 using the real-valued log-Mel inputs with different network sizes. Regularizing the spiking activity in SpikGRU during training allows us to adjust the trade-off between accuracy and number of operations, as shown in Fig. 2.A. We found that a small λ ($\in \{0.5, 1, 2\}$, depending on the topology) leads to a better accuracy than no regularization while reducing the number of operations. Therefore, SpikGRU with a small activity regularization achieves high accuracy (less than 0.5% below the ANN with similar accuracy) while demonstrating up to 82% reduction in number of operations, as shown in Table 1 and Fig. 2.C. This is achieved by using larger topologies than the ANN, but the number of operations is still lower due to the spike sparsity. Moreover, except for the smallest topology, SpikGRU with activity regularization achieves higher accuracy than the previous state-of-the-art SNN (94.5%, demonstrated by a spiking deep CNN in [10]), while requiring fewer operations (from -39% to -90%).

Spikes allows MACs to be replaced by ACs in the synaptic operations of spiking layers. However, due to the use of real-valued instead of spiking inputs, an important number of MACs remain in the feedforward connections of the first layer of SpikGRU. These MACs limit the efficiency of SNNs as they are not affected by the activity regularization. In particular, for small topologies with low activity rate, these operations become dominant (Fig 3.A). Therefore, in the next experiment, we evaluate the impact of using spiking instead of real-valued inputs.

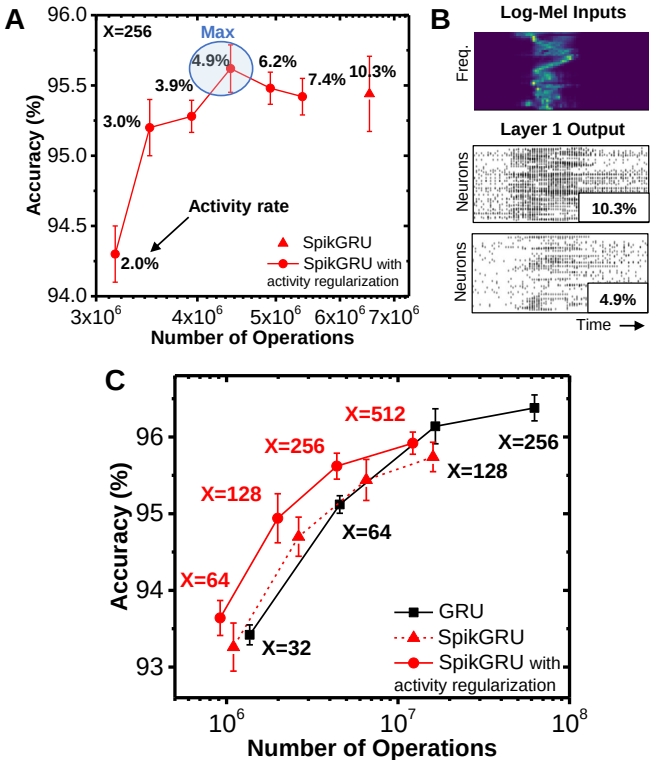

Fig. 2. **A.** Accuracy vs. number of operations per sample for SpikGRU (with size $X = 256$) with the different levels of activity regularization ($\lambda \in \{0.5, 1, 2, 4, 10, 50\}$). For each topology, the level of activity regularization leading to the highest accuracy is used for the results in Table 1 and Fig. 2.C. **B.** Output spikes from the first layer in SpikGRU with different activity rates. **C.** Accuracy vs. number of operations per sample for GRU and SpikGRU with different layer sizes (X), with and without activity regularization for SpikGRU.

Table 1. Accuracy and number of operations per sample on GSCD v2 for SpikGRU (with activity regularization) and GRU, and previous state-of-the-art SNN.

	Topo. (#Param.)	Accuracy (%)	#Ops
GRU	X=32 (14k)	93.4 ± 0.1	1.4M
	X=64 (46k)	95.1 ± 0.1	4.6M
	X=128 (165k)	96.1 ± 0.2	17M
	X=256 (625k)	96.4 ± 0.2	63M
SpikGRU	X=64 (31k)	93.6 ± 0.2	0.9M
	X=128 (111k)	94.9 ± 0.3	2.0M
	X=256 (418k)	95.6 ± 0.2	4.4M
	X=512 (1.6M)	95.9 ± 0.1	12M
SNN [10]	CNN (130k)	94.5	20M

3.2. Results with Spiking Inputs

In this second experiment, the KWS pipeline is used with the real-valued inputs converted to spikes. In this case, the input sparsity also impacts the accuracy and efficiency in SpikGRU. The input activity rate must be less than 1 to reduce the number of operations compared to using real-valued inputs (for which 1 MAC operation is performed at each timestep). The spiking inputs allow to replace the MACs due to the real-valued inputs by ACs (Fig 3.A). However, the reduction in number of operations is limited, as the input activity rate must be high enough to allow the input to be encoded with sufficient precision (otherwise the accuracy is largely degraded). Indeed, we observe that decreasing the input activity rate decreases the accuracy (Fig. 3.B).Therefore, the reduction in operations due to spiking inputs is not sufficient to compensate for the loss in accuracy. This leads to a degraded trade-off between accuracy and number of operations compared to the case of real-valued inputs (Fig. 3.C).

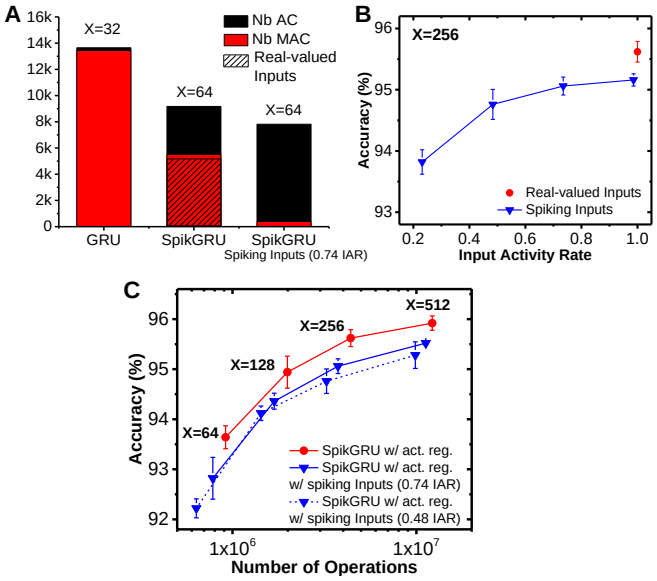

Fig. 3. **A.** Number of MAC and AC operations per timestep for GRU and SpikGRU (real-valued inputs) and SpikGRU with spiking inputs, with Input Activity Rate (IAR) of 0.74. MACs due to real-valued inputs in SpikGRU are shown in black. **B.** Accuracy vs. input activity rate with spiking inputs. **C.** Accuracy vs. number of operations per sample for SpikGRU with activity regularization (act. reg.) with real-valued or spiking inputs (IAR 0.74 and 0.48) with different hidden layer sizes (X).

This experiment shows that the precision of the input is crucial for the network accuracy. Therefore, adjusting the level of activity regularization and network size seems more efficient to decrease the number of operations with a minimal loss in accuracy. However, in hardware implementation, inputs are often quantized (and not real-valued) to increase the energy-efficiency. Therefore, there is also a trade-off between accuracy and energy consumption. Moreover, the hardware implementation of spike conversion from real-valued feature extractors induces an energy overhead. However, audio feature extractors consuming about 100 nW such as [17] output data in the form of spikes, making it possible to yield an ultra-low power end-to-end KWS solution. Therefore, SNNs offer two possibilities for the choice of the feature extractor: either using log-Mel features (quantized or not), or using a spiking feature extractor. In the first option, the first layer is trained to do the spike conversion, which allows minimal accuracy loss. The second option may be more advantageous if the extractor is able to produce relevant spiking features (leading the network to learn with a satisfactory accuracy) with sufficiently high sparsity. Besides, some sensors, such as the Dynamic Audio Sensor in [18], directly output spikes.

4. CONCLUSION

Our results have shown that recurrent SNNs, such as SpikGRU, are a promising solution for energy-efficient and accurate KWS. We have demonstrated the importance of regularizing the spiking activity, which allows SpikGRU to achieve a better trade-off between accuracy and energy-efficiency than GRU. Indeed, for the same accuracy, SpikGRU shows a lower number of operations. Moreover, the number of operations can be reduced by up to 82% compared to GRU with a loss of accuracy of less than 0.5%. We reach state-of-the-art results for SNNs on GSCD v2 with up to 95.9% accuracy with extremely low activity and low number of operations. Replacing the real-valued inputs by spiking inputs can further reduce the number of operations by exploiting data sparsity, but the accuracy-efficiency trade-off must be carefully considered. Therefore, SNNs offer the possibility of using either standard log-Mel feature extractors leading to high accuracy, or spiking feature extractors and dynamic sensors to increase the energy-efficiency. The main limitation of SNNs is the need for larger topologies compared to ANNs with the same accuracy, which increases the memory requirements. This could be mitigated by pruning synaptic connections and will be considered in future work.

5. REFERENCES

[1] Ivan Lopez-Espejo, Zheng-Hua Tan, John H. L. Hansen, and Jesper Jensen, "Deep spoken keyword spotting: An overview," *IEEE Access*, vol. 10, pp. 4169–4199, 2022.

[2] Yundong Zhang, Naveen Suda, Liangzhen Lai, and Vikas Chandra, "Hello edge: Keyword spotting on microcontrollers," 2017.

[3] Alice Coucke, Mohammed Chlieh, Thibault Gisselbrecht, David Leroy, Mathieu Poumeyrol, and Thibaut

Lavril, "Efficient keyword spotting using dilated convolutions and gating," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6351–6355.

[4] Mark Horowitz, "Computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014, pp. 10–14.

[5] Mike Davies, Andreas Wild, Garrick Orchard, Yulia Sandamirskaya, Gabriel A. Fonseca Guerra, Prasad Joshi, Philipp Plank, and Sumedh R. Risbud, "Advancing neuromorphic computing with loihi: A survey of results and outlook," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 911–934, 2021.

[6] Pete Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018.

[7] Peter Blouw and Chris Eliasmith, "Deep convolutional spiking neural networks for keyword spotting," in *Proceedings of Interspeech*, 2020, p. 2557–2561.

[8] Peter Blouw and Chris Eliasmith, "Event-driven signal processing with neuromorphic computing systems," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8534–8538.

[9] Jiadong Wang, Jibin Wu, Malu Zhang, Qi Liu, and Haizhou Li, "A hybrid learning framework for deep spiking neural networks with one-spike temporal coding," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8942–8946.

[10] Thomas Pellegrini, Romain Zimmer, and Timothée Masquelier, "Low-activity supervised convolutional spiking neural networks applied to speech commands recognition," in *IEEE Spoken Language Technology Workshop 2021*. Jan. 2021, Proc. 2021 IEEE Spoken Language Technology Workshop (SLT), pp. pp. 97–103, IEEE Xplore.

[11] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734, Association for Computational Linguistics.

[12] Kwantae Kim, Chang Gao, Rui Graça, Ilya Kiselev, Hoi-Jun Yoo, Tobi Delbruck, and Shih-Chii Liu, "A 23μw solar-powered keyword-spotting asic with ring-oscillator-based time-domain feature extraction," in *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, 2022, vol. 65, pp. 1–3.

[13] Manon Dampfhoffer, Thomas Mesquida, Alexandre Valentian, and Lorena Anghel, "Investigating current-based and gating approaches for accurate and energy-efficient spiking recurrent neural networks," in *Artificial Neural Networks and Machine Learning – ICANN 2022. Lecture Notes in Computer Science, vol 13531*, 2022.

[14] Manon Dampfhoffer, Thomas Mesquida, Alexandre Valentian, and Lorena Anghel, "Are SNNs really more energy-efficient than ANNs? An in-depth hardware-aware study," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2022.

[15] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proceedings of Interspeech*, 2019.

[16] Emre Neftci, Hesham Mostafa, and Friedemann Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Processing Magazine*, vol. 36, pp. 51–63, 11 2019.

[17] Dewei Wang, Sung Justin Kim, Minhao Yang, Aurel A. Lazar, and Mingoo Seok, "9.9 a background-noise and process-variation-tolerant 109nw acoustic feature extractor based on spike-domain divisive-energy normalization for an always-on keyword spotting device," in *2021 IEEE International Solid- State Circuits Conference (ISSCC)*, 2021, vol. 64, pp. 160–162.

[18] Jithendar Anumula, Daniel Neil, Tobi Delbruck, and Shih-Chii Liu, "Feature representations for neuromorphic audio spike streams," *Frontiers in Neuroscience*, vol. 12, 2018.