T-NGA: TEMPORAL NETWORK GRAFTING ALGORITHM FOR LEARNING TO PROCESS SPIKING AUDIO SENSOR EVENTS

Shu Wang[†]   Yuhuang Hu[†]   Shih-Chii Liu

Institute of Neuroinformatics, University of Zürich and ETH Zürich, Zürich, Switzerland

ABSTRACT

Spiking silicon cochlea sensors encode sound as an asynchronous stream of spikes from different frequency channels. The lack of labeled training datasets for spiking cochleas makes it difficult to train deep neural networks on the outputs of these sensors. This work proposes a self-supervised method called Temporal Network Grafting Algorithm (T-NGA), which grafts a recurrent network pretrained on spectrogram features so that the network works with the cochlea event features. T-NGA training requires only temporally aligned audio spectrograms and event features. Our experiments show that the accuracy of the grafted network was similar to the accuracy of a supervised network trained from scratch on a speech recognition task using events from a software spiking cochlea model. Despite the circuit non-idealities of the spiking silicon cochlea, the grafted network accuracy on the silicon cochlea spike recordings was only about 5% lower than the supervised network accuracy using the N-TIDIGITS18 dataset. T-NGA can train networks to process spiking audio sensor events in the absence of large labeled spike datasets.

Index Terms— spiking silicon cochlea sensor, event-driven audio processing, deep neural network, self-supervised learning, speech recognition

1. INTRODUCTION

With the increasing use of speech command interfaces on mobile devices, finding lightweight networks for edge audio tasks, *e.g.*, Voice Activity Detection (VAD), Speaker Verification (SV), and Keyword Spotting (KWS), has become a fast growing research area [1]. Recent methods to produce low-memory footprint or low-compute networks include the use of delta activation update [2] and Neural Architectural Search [3] methods.

The inputs to these networks are typically spectrogram features extracted from the audio samples, *e.g.*, Mel-Frequency Cepstral Coefficients (MFCCs) and Log-Mel spectrograms. On the other hand, spiking silicon cochlea sensors, which implement a simple model of the biological cochlea, encode audio as an asynchronous stream of output spikes from a set of frequency channels. The spiking Dynamic Audio Sensor (DAS) incorporates filter circuits that model the frequency selectivity of the basilar membrane, the rectification of inner hair cells, and the spike generation of the auditory nerve fibers [4, 5]. The DAS output events have been used to drive low-latency localization solutions together with the separation of the spike streams produced by competing talkers [6]. These studies show that it is possible to simultaneously localize multiple speakers, to separate their spike streams, and to estimate the speech envelope of an individual
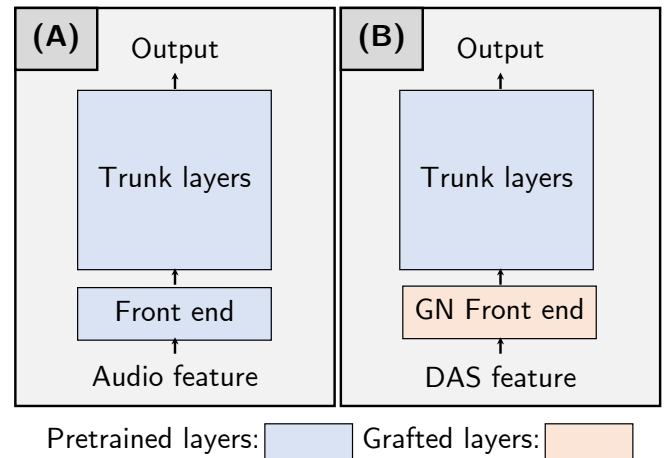
Fig. 1. Temporal Network Grafting Algorithm (T-NGA). "GN" stands for grafted network.

speaker from the separated spike streams. The DAS output has been used for edge audio tasks such as SV [7] and digit recognition [2] where the low-power and low-latency properties of an edge audio device are needed. Other methods use front-end software models that produce audio events, which are then processed by a spiking neural network [8, 9, 10]. Custom edge Application-Specific Integrated Circuit (ASIC) designs that include the VAD chip by [11] use a spiking cochlea front-end together with a multilayer binary perceptron, and demonstrate a low power consumption of $1\,\mu$W.

Training deep neural networks for audio processing often requires large labeled datasets such as LibriSpeech which contains over 960 hours of transcribed audio [12]. There is a lack of large labeled spike datasets for event-driven sensors, thereby preventing the more extensive use of deep networks for processing spiking cochlea output events. To mitigate the expensive costs for labeling spiking audio datasets, we propose a self-supervised Temporal Network Grafting Algorithm (T-NGA), which requires only temporally aligned audio waveforms and spike recordings for training. In other words, T-NGA training does not require labeled DAS recordings. In this work, we show how T-NGA can exploit a pretrained network that processes audio spectrogram features to produce a *grafted network* (GN) that processes event features via Transfer Learning. T-NGA extends the Network Grafting Algorithm (NGA) [13] which was originally proposed for vision modalities to temporal audio sequences.

We studied and validated T-NGA using a Recurrent Neural Network (RNN) on a simple Automatic Speech Recognition (ASR) task (Section 3), first using spike recordings generated from an ideal soft-

ware spiking cochlea model and then from the DAS recordings in the N-TIDIGITS18 dataset [14]. We investigated whether the accuracy of the GN obtained from T-NGA training could achieve the accuracy of a network trained from scratch using the labeled spiking cochlea recordings; and how the silicon circuit non-idealities and recording non-idealities affect the accuracy of GN. The contributions of this work can be summarized as follows:

- We propose a self-supervised training method, T-NGA, which avoids the need of expensive labeling when training on novel spiking cochlea events for an acoustic task such as ASR.

- T-NGA, to our knowledge, is the first algorithm that utilizes temporally aligned audio spectrogram and event features for training a network that uses spiking cochlea events for an ASR task.

2. METHODS

2.1. Temporal Network Grafting Algorithm (T-NGA)

Fig. 1 illustrates the working principle of T-NGA. An RNN $\mathcal{F}(\cdot)$ is pretrained on spectrogram features extracted from the audio waveform samples. The pretrained network $\mathcal{F}(\cdot)$ consists of two parts: a front end $\mathcal{F}_{\text{front}}$ and trunk layers $\mathcal{F}_{\text{trunk}}$ (Fig. 1 (A)). The weights of the pretrained network are frozen during the T-NGA training. After training, a newly-trained grafted front end $\mathcal{G}_{\text{front}}(\cdot)$ together with the pretrained trunk layers $\mathcal{F}_{\text{trunk}}$ form a *grafted network* (GN) that processes event features computed from the spiking cochlea events (Fig. 1 (B)).

Let $\{(\mathbf{a}_i, t_i^a)\}_{i=1}^{T^a}$ and $\{(\mathbf{s}_t, t_i^s)\}_{i=1}^{T^s}$ be the temporally aligned spectrogram and event features where t_i^a and t_i^s are timestamps of $\mathbf{a}_i$ and $\mathbf{s}_i$ respectively. Note that T^a may not be equal to T^s depending on the choice of feature configuration (see Section 2.3).

$\mathcal{F}(\cdot)$ processes the spectrogram features $\{(\mathbf{a}_i, t_i^a)\}_{i=1}^{T^a}$ and returns the prediction $\{y_l\}_{l=1}^{L^a}$:

$$\{\mathbf{h}_i\} = \mathcal{F}_{\text{front}}(\{\mathbf{a}_i\}), \qquad \{y_l\} = \mathcal{F}_{\text{trunk}}(\{\mathbf{h}_i\}) \tag{1}$$

where $\{(\mathbf{h}_i, t_i^a)\}_{i=1}^{T^a}$ are the pretrained states, *i.e.*, hidden states of $\mathcal{F}_{\text{front}}$. Similarly, GN processes event features $\{(\mathbf{s}_t, t_i^s)\}_{i=1}^{T^s}$ and outputs prediction $\{y_l^s\}_{l=1}^{L^s}$:

$$\{\mathbf{g}_i\} = \mathcal{G}_{\text{front}}(\{\mathbf{s}_i\}), \qquad \{y_l^s\} = \mathcal{F}_{\text{trunk}}(\{\mathbf{g}_i\}) \tag{2}$$

where $\{(\mathbf{g}_i, t_i^s)\}_{i=1}^{T^s}$ are the grafted states, *i.e.*, hidden states of $\mathcal{G}_{\text{front}}$. The dimension of $\mathbf{g}_i$ is the same as the dimension of $\mathbf{h}_i$.

T-NGA training maximizes the similarity between $\{\mathbf{h}_i\}$ and $\{\mathbf{g}_i\}$. When $T^a \neq T^s$, we select T features from both $\{\mathbf{h}_i\}$ and $\{\mathbf{g}_i\}$ where $T \leq \min(T^a, T^s)$ and timestamps of the corresponding states t_i^a and t_i^s are the closest in time. T-NGA training optimizes the following loss function:

$$\mathcal{L} = \text{NCS}(\{\mathbf{h}_i\}, \{\mathbf{g}_i\}) + \text{MAE}(\{\mathbf{h}_i\}, \{\mathbf{g}_i\}) \tag{3}$$

$$= 1 + \frac{1}{T}\sum_{i=1}^{T} \frac{\mathbf{h}_i \cdot \mathbf{g}_i}{\|\mathbf{h}_i\| \cdot \|\mathbf{g_i}\|} + \frac{1}{T}\sum_{i=1}^{T} |\mathbf{h}_i - \mathbf{g}_i| \tag{4}$$

where $\text{NCS}(\cdot, \cdot)$ and $\text{MAE}(\cdot, \cdot)$ compute the negative cosine similarity and the mean absolute error between aligned states, respectively. When the training is completed, the grafted front end $\mathcal{G}_{\text{front}}$ encodes the event features into a set of grafted states that can be directly used by the pretrained trunk layers $\mathcal{F}_{\text{trunk}}$.
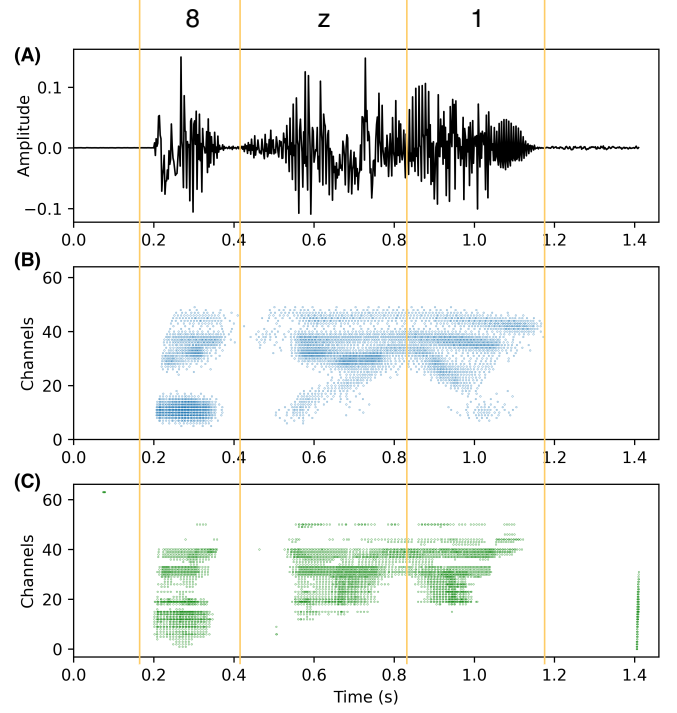

Fig. 2. Time waveform **(A)** of a sample "8z1" from the TIDIGITS dataset. Also shown are spikes generated using an ideal software cochlea model **(B)**, and real DAS spikes **(C)** from the 64 cochlea frequency channels in response to the audio sample.

2.2. Spiking Cochlea

The Dynamic Audio Sensor (DAS) has two independent 64-stage cascaded filter banks driven by two microphone inputs. The frequency selectivity of the cochlea channels ranges roughly from 50 Hz to 20 kHz on a log spacing across the channels. The DAS has been captured in several cochlea designs [5, 4, 15]. The spiking cochlea designed in [15] was used to generate the spike recordings of the TIDIGITS dataset to create a spiking cochlea dataset called N-TIDIGITS18. Details of this spiking dataset are given in Sec 2.5.

In this work, we also use a software model of a frequency channel which includes a second-order filter, a half-wave rectifier, and a linear leak integrate-and-fire neuron model used in the design reported in [15]. The transfer function of the n-th filter in a cascaded set of second-order filters is given by

$$H_n(s) = \prod_{i=1}^{n} \left(\frac{\tau_i\, s}{\tau_i^2\, s^2 + \tau_i\, s/Q + 1} \right) \tag{5}$$

where $\tau_i = \frac{1}{2 \cdot \pi \cdot f_i}$, f_i is the center frequency of the i-th channel, and Q is the filter quality factor. The half-wave rectification is described as:

$$v_r = \max(0, V_n - V_{\text{ref}}) \tag{6}$$

where v_r is the rectifier output used to drive the neuron, V_n is the filter output of the channel and V_{ref} is a constant. The parameters of the model are set so that the spike responses of the channels are similar to those of the hardware spiking cochlea.

An example of the spikes generated from the software spiking cochlea model and the spikes recorded with the hardware spiking

3274

cochlea in response to the sample "8z1" (8Z1A.WAV from speaker AE in TIDIGITS) is shown in Fig. 2.
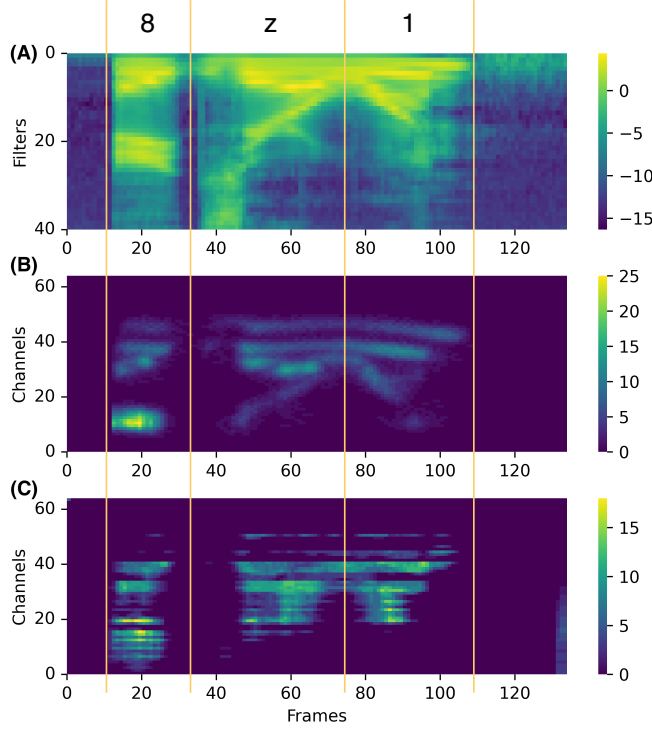

Fig. 3. Features extracted from the audio sample "8z1". Log-Mel spectrogram features **(A)**. TBSC features generated from simulated spikes **(B)** and DAS responses **(C)** using a configuration 25w/10s.

2.3. Feature Extraction

Features are extracted from both the audio and spike samples. For audio samples, we computed the Log-Mel spectrogram features using a 25 ms window, 10 ms stride, and 40 filter bank channels as shown in Fig. 3 (A). For the spike recordings, we used the time-binned spike count (TBSC) method [14] to convert spikes into event features. Assume that the spike stream consists of N spikes $\{e_i = (t_i, f_i)\}_{i=1}^{N}$ where timestamps t_i are in ascending order. The TBSC feature for a time window ΔW is then defined as $\mathbf{s}_j^{(f)} = \text{card}(\{e_i | t_i \in j\text{-th window}, f_i = f\})$ where $\mathbf{s}_j^{(f)}$ represents TBSC feature value at j-th window for channel frequency f; $\text{card}(\cdot)$ is the cardinality of selected events. ΔW is 10 ms or 25 ms and the stride is 10 ms for the TBSC features used in the study. Examples of TBSC features with $\Delta W = 25$ ms are shown in Fig. 3 (B) and (C). For convenience, feature configurations are coded as window/stride, *e.g.*, 25w/10s means 25 ms window and 10 ms stride.

2.4. Network Architecture

The RNN used in this work consists of two Gated Recurrent Unit (GRU) [16] layers with 256 units each followed by a fully connected layer of 200 units with a LeakyReLU activation [17], and a final classification layer of 12 output units. The network has a total of 677 K parameters.

2.5. Datasets

We used the men and women speaker recordings in the TIDIGITS dataset [18] to match the spike recordings in the N-TIDIGITS18 dataset [14]. N-TIDIGITS18 consists of 8,623 training samples and 8,700 testing samples. Each sample consists of a single digit or multiple spoken digits. In total, there are 11 digits ("o", "zero" and "1" to "9").

For T-NGA training, the N-TIDIGITS18 recordings and the corresponding audio samples were aligned temporally using the Dynamic Time Warping (DTW) algorithm [19] before feature extraction. The spike recordings generated by the software spiking cochlea model are already aligned with the corresponding audio samples.

3. RESULTS

We conducted experiments on a speech recognition task using the datasets described in Section 2.5. All networks in this section were trained for 50 epochs using the Adam optimizer [20]. For the supervised training experiments, we used a learning rate of 3e-4 and the Connectionist Temporal Classification (CTC) loss function [21]. For T-NGA, we used a learning rate of 1e-3 and the T-NGA loss function in Eq. 4. Reported results show the mean and standard deviation of the word error rate (WER) for 5 runs of each experiment.

3.1. T-NGA experiments

We first trained a network `PT-25` on TIDIGITS using spectrogram features as shown in Table 1(A). The `PT-25` model was subsequently used as the pretrained model for T-NGA training to obtain grafted networks (GNs). For comparison, we also trained supervised networks (SNs) through standard supervised training on spiking cochlea datasets using event features.

Table 1(B) shows the results of using the event features created from ideal software spiking cochlea recordings. The GNs `BGN-25`, `BGN-10` reached similar or better WERs than the SNs `B-25`, `B-10`. These result show that T-NGA training could be advantageous as it does not require a labeled event dataset and still produces a network that shows equivalent accuracy.

Results on N-TIDIGITS18 presented in Table 1(C) show that the SNs `S-25`, `S-10` achieved ∼10% WER while the GNs `GN-25`, `GN-10` reached ∼15% WER. The higher WER of the GNs compared to the corresponding SNs is because the hardware spiking cochlea includes circuit non-idealites that affect the distribution of the output cochlea spikes, and the temporal mismatch between the audio samples and event sample still exists after DTW alignment. In Section 3.2, we show that the WER of GN increased when training on spike recordings generated by the same software cochlea model which includes circuit non-idealities.

Table 1 also shows that both GNs and SNs perform better when using a 10w/10s event feature configuration, *i.e.*, 10w/10s characterizes the spike recordings better than 25w/10s. T-NGA training for `BGN-10` and `GN-10` models used 25w/10s Log-Mel spectrogram features and 10w/10s TBSC features. The results show that T-NGA can maximize the similarity between temporally aligned grafted states and pretrained states successfully while the feature configurations for the two modalities are different.

3.2. Analysis

We present two additional experiments to better understand the results obtained from T-NGA training and the resulting GN.

Table 1. Comparison of T-NGA with supervised training on an ASR task. Log-Mel spectrogram features are generated with a 25w/10s configuration. TBSC features are generated with 25w/10s or 10w/10s configurations. T-NGA results are highlighted in blue.

Model	Training Method	Features	Feature Configuration	WER (%)
(A) on TIDIGITS dataset				
PT-25	Supervised	Log-Mel	25w/10s	0.80±0.15
(B) ideal software cochlea model on TIDIGITS dataset				
B-25	Supervised	SW-TBSC†	25w/10s	2.20±0.36
B-10	Supervised	SW-TBSC	10w/10s	1.70±0.17
BGN-25	T-NGA	Log-Mel+SW-TBSC	25w/10s	1.96±0.05
BGN-10	T-NGA	Log-Mel+SW-TBSC	10w/10s	1.70±0.04
(C) on N-TIDIGITS18 dataset				
S-25	Supervised	TBSC	25w/10s	10.18±0.39
S-10	Supervised	TBSC	10w/10s	9.63±0.20
Ref. [14]	Supervised	TBSC	n/a	13.9%
GN-25	T-NGA	Log-Mel+TBSC	25w/10s	15.01±0.82
GN-10	T-NGA	Log-Mel+TBSC	10w/10s	14.85±0.47

†SW: Software cochlea model; TBSC: Time-binned spike count features.

We first investigated quantitatively whether the circuit non-idealities of a spiking cochlea affect the performance of GN and SN. We generated another event dataset using the same software spiking cochlea model that includes circuit non-idealities from the the neuron spiking threshold, and filter quality factor variation. As a result, the WER of the GN was ∼3% higher than the SN as shown in Table 2. This result reveals that these non-idealities in the hardware spiking cochlea contributed to the gap in WER between GNs and SNs as observed in Table 1(C).

Table 2. T-NGA and supervised training results using a software spiking cochlear model that includes circuit non-idealities. Feature configuration is 25w/10s.

Training Method	Features	WER (%)
Supervised	SW-TBSC	5.31±1.07
T-NGA	Log-Mel+SW-TBSC	8.30±0.18

We also explored the encoded content in the grafted states of GN by decoding a given set of grafted states through the pretrained front end by following [13]. The outcome of this decoding procedure is a decoded Log-Mel spectrogram that represents the given grafted states. The optimization of this decoding procedure was carried out for 5,000 iterations using the Adam optimizer and a learning rate of 1e-2. To suppress the noise in the decoded Log-Mel spectrogram, we clipped the amplitude at -10. Fig. 4 shows a Log-Mel spectrogram that was decoded from the grafted states that were trained on a N-TIDIGITS18 sample. We found that the trained GN front end can generate the grafted features which often correctly represents the input audio signal. The decoded spectrogram shown in Fig. 4 (B) is visually similar to the original Log-Mel spectrogram shown in Fig. 4 (A). Fig. 4 (C) illustrates that the distributions of the amplitudes for the original and decoded spectrograms match well.

4. CONCLUSION

We propose a self-supervised method called Temporal Network Grafting Algorithm (T-NGA) which exploits a RNN trained on audio spectrogram features to process event features through a newly trained grafted front end. This algorithm mitigates the need for
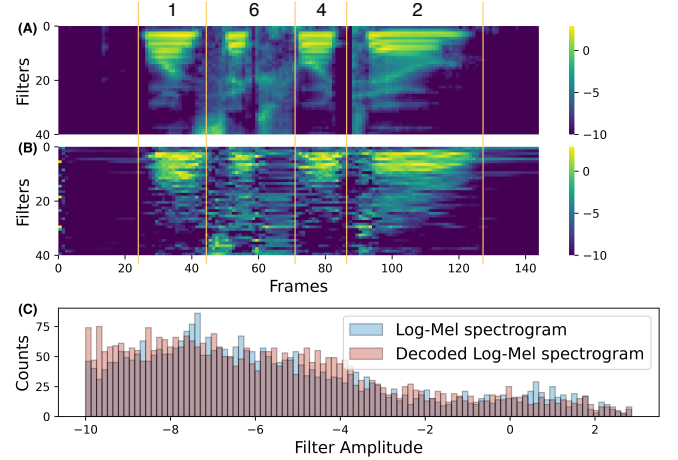

Fig. 4. Log-Mel spectrogram features **(A)** computed from the sample "1-6-4-2". The decoded Log-Mel spectrogram features **(B)** are qualitatively similar to the features in **(A)**. The histogram **(C)** shows the distribution of filter amplitudes for the two feature sets.

labeled spike recordings and only requires temporally aligned audio and spike samples.

We demonstrate T-NGA on a speech recognition task using TIDIGITS and N-TIDIGITS18 datasets; and show that the accuracy of the GN is equivalent to that of a network trained from scratch on the labeled software spiking cochlea recordings. When validated on real DAS recordings, the accuracy of the GN was only ∼5% lower than the accuracy of the SN despite the silicon circuit non-idealities reflected in the DAS recordings. The T-NGA works well even when the feature configuration of the Log-Mel spectrogram is different from that of the event features as shown in Section 3. Through self-supervised training, GN produces grafted states that are similar to the pretrained states as shown by the decoded spectrogram features.

Future work includes testing on a DAS dataset, which already includes the synchronized audio and spike recordings, and testing on datasets used for edge audio tasks such as keyword spotting.

5. REFERENCES

[1] R. Prabhavalkar, R. Alvarez, C. Parada, P. Nakkiran, and T. N. Sainath, "Automatic gain control and multi-style training for robust small-footprint keyword spotting with deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4704–4708.

[2] C. Gao, S. Braun, I. Kiselev, J. Anumula, T. Delbruck, and S. Liu, "Real-time speech recognition for IoT purpose using a delta recurrent neural network accelerator," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2019, pp. 1–5.

[3] B. Zhang, W. Li, Q. Li, W. Zhuang, X. Chu, and Y. Wang, "Autokws: Keyword spotting with differentiable architecture search," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 2830–2834.

[4] S.-C. Liu, A. van Schaik, B. A. Minch, and T. Delbruck, "Asynchronous binaural spatial audition sensor with $2 \times 64 \times 4$ channel output," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 4, pp. 453–464, 2014.

[5] M. Yang, C.-H. Chien, T. Delbruck, and S.-C. Liu, "A 0.5 V 55 μW 64×2 channel binaural silicon cochlea for event-driven stereo-audio sensing," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 11, pp. 2554–2569, 2016.

[6] J. Anumula, E. Ceolini, Z. He, A. Huber, and S-C. Liu, "An event-driven probabilistic model of sound source localization using cochlea spikes," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018.

[7] E. Ceolini, J. Anumula, S. Braun, and S.-C. Liu, "Event-driven pipeline for low-latency low-compute keyword spotting and speaker verification system," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7953–7957.

[8] J. Wu, E. Yılmaz, M. Zhang, H. Li, and K. C. Tan, "Deep spiking neural networks for large vocabulary automatic speech recognition," *Frontiers in neuroscience*, vol. 14, pp. 199, 2020.

[9] E. Yılmaz, O. B. Gevrek, J. Wu, Y. Chen, X. Meng, and H. Li, "Deep convolutional spiking neural networks for keyword spotting," in *Proceedings of Interspeech*, 2020, pp. 2557–2561.

[10] F. Martinelli, G. Dellaferrera, P. Mainar, and M. Cernak, "Spiking neural networks trained with backpropagation for low power neuromorphic implementation of voice activity detection," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8544–8548.

[11] M. Yang, C. Yeh, Y. Zhou, J. P. Cerqueira, A. A. Lazar, and M. Seok, "A 1μW voice activity detector using analog feature extraction and digital deep neural network," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, Feb 2018, pp. 346–348.

[12] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[13] Y. Hu, T. Delbruck, and S-C. Liu, "Learning to exploit multiple vision modalities by using grafted networks," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J-M. Frahm, Eds., Cham, 2020, pp. 85–101, Springer International Publishing.

[14] J. Anumula, D. Neil, T. Delbruck, and S.-C. Liu, "Feature representations for neuromorphic audio spike streams," *Frontiers in Neuroscience*, vol. 12, pp. 23, 2018.

[15] V. Chan, S.-C. Liu, and A. van Schaik, "AER EAR: A matched silicon cochlea pair with address event representation interface," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 1, pp. 48–59, 2007.

[16] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. 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.

[17] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[18] R. G. Leonard and G. R. Doddington, "Tidigits ldc93s10," 1993.

[19] T. Giorgino, "Computing and visualizing dynamic time warping alignments in R: The dtw package," *Journal of Statistical Software, Articles*, vol. 31, no. 7, pp. 1–24, 2009.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun, Eds., 2015.

[21] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 369–376.