Neuromorphic Lip-Reading with Signed Spiking Gated Recurrent Units

Manon Dampfhoffer
Univ. Grenoble Alpes, CEA, List,
F-38000 Grenoble, France
manon.dampfhoffer@cea.f

Thomas Mesquida
Univ. Grenoble Alpes, CEA, List,
F-38000 Grenoble, France
thomas.mesquida@cea.fr

Abstract

Automatic Lip-Reading (ALR) requires the recognition of spoken words based on a visual recording of the speaker's lips, without access to the sound. ALR with neuromorphic event-based vision sensors, instead of traditional frame-based cameras, is particularly promising for edge applications due to their high temporal resolution, low power consumption and robustness. Neuromorphic models, such as Spiking Neural Networks (SNNs), encode information using events and are naturally compatible with such data. The sparse and event-based nature of both the sensor data and SNN activations can be leveraged in an end-to-end neuromorphic hardware pipeline for low-power and low-latency edge applications. However, the accuracy of SNNs is often largely degraded compared to state-of-the-art non-spiking Artificial Neural Networks (ANNs). In this work, a new SNN model, the Signed Spiking Gated Recurrent Unit (SpikGRU2+), is proposed and used as a task head for event-based ALR. The SNN architecture is as accurate as its ANN equivalent, and outperforms the state-of-the-art on the DVS-Lip dataset. Notably, the accuracy is improved by 25% (respectively 4%) compared to the previous state-of-the-art SNN (respectively ANN). In addition, the SNN spike sparsity can be optimized to further reduce the number of operations up to 22x compared to the ANN while maintaining a high accuracy. This work opens up new perspectives for the use of SNNs for accurate and low-power end-to-end neuromorphic gesture recognition. Code is available[1].

1. Introduction

Automatic Lip-Reading (ALR), also called Visual Speech Recognition (VSR), aims at recognizing speech only based on the vision of the speaker's lip movements. ALR can be used in addition to audio-based speech recognition, for instance to improve speech recognition in noisy environments [34], or alone when no audio is available. ALR has many real-life applications, ranging from improved hearing aids [40] to video surveillance. Deep learning methods have shown impressive abilities to perform audio and visual speech recognition [29, 33]. However, they are also very resource-intensive, requiring large memory and power consumption, which makes them difficult to embed on low-resources tiny edge devices. This limits their use on portable devices, such as hearing aids.

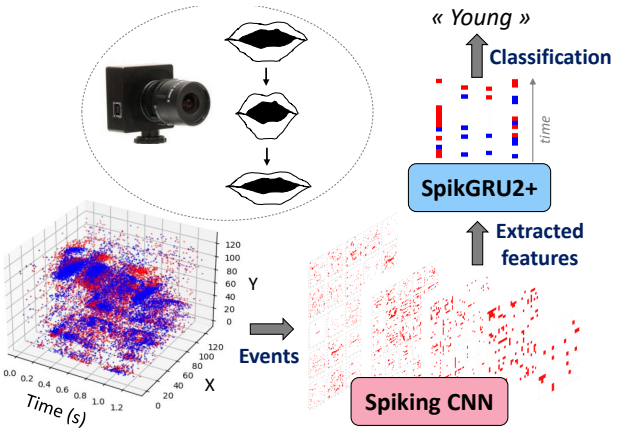

Figure 1. End-to-end neuromorphic lip-reading. Event data captured by an event-based camera are fed to a Spiking Neural Network composed of a Spiking CNN feature extractor and SpikGRU2+ as task head.

Neuromorphic computing promises higher efficiency, by taking inspiration from biological cognitive systems [30]. Neuromorphic sensors, such as event-based cameras (also called Dynamic Vision Sensors) have been proposed as low-power alternatives to traditional frame-based cameras [26]. Indeed, event-based cameras asynchronously produce polarized events when a local change in intensity is detected. Neuromorphic algorithms and hardware, such as Spiking Neural Networks (SNNs), can fully benefit from the event-based nature of neuromorphic sensors in an end-to-end event-based pipeline, promising very high efficiency. In-

[1] https://github.com/manondampfhoffer/SpikGRU-DVSLip

This publication was made possible by the use of the FactoryIA supercomputer, financially supported by the Ile-De-France Regional Council.

spired by how biological neurons use electrical pulses to transmit information through the synapses, SNNs computations are based on asynchronous accumulations of sparse spike events, which can be efficiently leveraged in low-power neuromorphic hardware [15, 31]. In addition, as they share the same data format, SNNs can directly handle data from neuromorphic sensors in an end-to-end event-based implementation. SNNs have demonstrated promising performance with event-based data in optical flow prediction [4, 8, 21], gesture recognition [1, 5, 38] and audio speech recognition [11, 46]. However, training deep SNNs with gradient descent is challenging due to the spiking activations, that are sparse and of low resolution compared to standard Artificial Neural Networks (ANNs) [14, 37]. The accuracy of SNNs is usually degraded compared to an ANN with the same topology, and hence the use of SNNs in practical applications is often discarded [9].

In this work, we address the challenge of end-to-end neuromorphic lip-reading, ie. performing the ALR task from event-based camera data with a SNN (see Fig. 1). The main contributions are summarized as follows:

- A new SNN model, called Signed Spiking Gated Recurrent Unit (SpikGRU2+), is provided and used as task head for event-based ALR.
- An effective data augmentation for spatio-temporal event-based data is proposed.
- The SNN yields 25% higher accuracy than the previous SNN state-of-the-art, and even 4% higher accuracy than the previous ANN state-of-the-art on the challenging DVS-Lip dataset [43].
- Due to the high spike sparsity, which is enhanced by optimization, the SNN can reduce up to 22x the number of operations compared to its ANN equivalent, while maintaining a high accuracy.

2. Related Works

2.1. Spiking Neural Networks

Leaky Integrate-and-Fire Models

Deep SNNs are based on a time-discretized version of the Leaky-Integrate-and-Fire (LIF) neuron model, which describes the dynamic of the membrane potential of neurons and the spike firing [14, 37]:

$$v_t^l = \beta \odot v_{t-1}^l + W^l s_t^{l-1} + b^l - V_{th} s_{t-1}^l \tag{1}$$

$$s_t^l = SpikeAct(v_t^l) \tag{2}$$

v_t^l and s_t^l are respectively the membrane potential and output spikes of neurons from layer l at time t. The leak $\beta \in [0, 1]$ determines the exponential decay of the membrane potential with time. W and b represent the weights and biases parameters of the SNN layer. $\odot$ denotes element-wise multiplication. According to the spiking activation function (SpikeAct), the neuron fires a spike when the membrane potential is superior to the threshold V_{th}, after which V_{th} is substracted from the membrane potential:

$$SpikeAct(x) = \begin{cases} 1, & \text{if } x \geq V_{th} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Few works [24, 44] have considered using signed spiking neurons, which can fire both positive and negative spikes, instead of the traditional spiking activation function. This was motivated by the use of ANN-to-SNN conversion training (which maps a trained ANN onto a SNN). Indeed, in that case, negative spikes can allow to compensate for an excess of positive spikes fired (due to the asynchronous nature of spike firing) so that the total activations of the SNN better match the one of the equivalent ANN. The signed spiking activation function can be defined as:

$$SpikeAct_{signed}(x) = \begin{cases} 1, & \text{if } x \geq V_{th} \\ -1, & \text{if } x \leq -V_{th} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Spiking Gated Recurrent Units

SNNs are particularly suited for processing spatio-temporal data, due to their inherent spatio-temporal dynamics. Rich spiking neuron models can allow to capture spatio-temporal patterns, using one or several stateful variables [16] with learnable time constants [18], or learnable synaptic [48] or axonal [42] delays. However, these spiking models still lag behind state-of-the-art recurrent architectures on difficult tasks [3]. A spiking adaptation of the Long Short-Term Memory network (LSTM) was proposed in [28], where each activation function (including the gates) in the cell model is replaced with a spiking one. However, this largely degrades the accuracy compared to the original ANN Gated Recurrent Unit (GRU), as shown in [3]. Another proposition of spiking GRU (SpikGRU) was introduced in [11], which only uses a spiking activation function at the output of the cell. Therefore, the candidate state (input current) i and the hidden state (membrane potential) v are computed in full precision, which has been shown to improve accuracy while inducing negligible overhead [11]. The SpikGRU model is defined with a single gate (z), as follows:

$$z_t^l = \sigma(W_z s_t^{l-1} + U_z s_{t-1}^l + b_z) \tag{5}$$

$$i_t^l = \alpha \odot i_{t-1}^l + W_i s_t^{l-1} + U_i s_{t-1}^l + b_i \tag{6}$$

$$v_t^l = z_t^l \odot v_{t-1}^l + (1 - z_t^l) \odot i_t^l - V_{th} s_{t-1}^l \tag{7}$$

$$s_t^l = SpikeAct(v_t^l) \tag{8}$$

W_i, W_z and U_i, U_z being the weight matrices of feed-forward and recurrent connections respectively, b_i and b_z the biases, σ the sigmoid function, and $\alpha \in [0, 1]$ determines the decay rate of current i.

2.2. Event-based Lip-Reading

Lip-Reading Datasets

ALR methods aim at recognizing alphabets, digits, words or sentences based on video recordings of the speaker's lips. Most public datasets [7, 39, 45] are recorded with conventional cameras. Recently, the first event-based lip-reading dataset (DVS-Lip [43]), recorded with an event-based camera, has been released. In event-based cameras [26], pixels are sensitive to local changes in intensity, and asynchronously produce an event if a change in brightness occurs, or remain silent otherwise. Events have a polarity (positive or negative), indicating the direction of the change. Compared with standard camera with a fixed frame rate, event-based cameras have a higher temporal resolution (in the order of the microsecond) [26], which makes them attractive for the ALR task requiring to detect finer-grained spatio-temporal patterns [43]. Moreover, they are low-power and robust to challenging lighting conditions.

Lip-Reading Methods

Popular ALR deep learning methods are based on a frontend with convolutional layers that extracts the spatial features, and a backend with recurrent layers that processes the spatio-temporal information [19, 41, 43]. In particular, ResNet architectures for the frontend and bi-directional GRU layers for the backend have shown high accuracy on standard lip-reading datasets [19, 43], as well as the event-based (DVS-Lip) one [43]. Smaller-scale models have also been proposed, using Graph Neural Networks for the frontend [32], or using reservoirs for the backend [47]. SNNs, due to their event-based nature, are particularly appealing for event-based data, allowing a low-power end-to-end event-based pipeline [1, 5, 38]. Recently, the challenging event-based lip-reading task has been addressed with a SNN inspired by the Multi-grained Spatio-Temporal features Perceived network architecture (MSTP) [43], called Spiking MSTP [3]. The frontend is a ResNet-18 based on Spiking Element Wise (SEW-ResNet) layers [17]. Several backends are experimented, such as a spiking adaptation of GRU from [28] and stateful synapses [16], but no satisfying solution is found, as the SNN accuracy is largely below the one of the ANN MSTP. This is explained by the poor performance of the spiking backend.

2.3. Event-based Data Augmentation

Due to the limited size of event-based datasets, models can largely benefit from data augmentation to improve generalization [25]. Previous works [2, 20, 25] have investigated data augmentation techniques for event-based data, mostly geometrical data augmentation techniques inspired by image data augmentation, such as cropping, flipping, rolling, rotation, shear, etc. [20] have proposed event dropping in space (equivalent to image cutout, i.e. removing events in a 2D area), time (removing events in a time interval), and random dropping of events. Temporal data augmentation techniques have been less explored (except for event dropping in time within a single interval), while we believe it is crucial for ALR data with spatio-temporal patterns.

3. Methods

3.1. Signed Spiking Gated Recurrent Unit

The proposed model is inpired by SpikGRU [11], a spiking adaptation of GRU, with two significant improvements: (1) the use of the signed spiking activation function to replace the standard spiking activation function and, (2) the addition of a second gate in the neuron cell model. This model is called Signed Spiking Gated Recurrent Unit (SpikGRU2+, '2' stands for the second gate and '+' stands for the signed activation function).

As mentioned in Section 2, Signed SNNs have been considered in the context of ANN-to-SNN conversion training to help the SNN matching the activations of the trained ANN. However, we argue that using Signed SNN can also improve the accuracy of directly trained SNNs. In particular, the signed spiking activation function (SpikeAct$_{signed}$) more closely resembles the hyperbolic tangent function ($tanh$), which is typically the activation function used in an ANN GRU [6]. Therefore, SpikeAct$_{signed}$ seems more relevant than the standard SpikeAct for a spiking GRU. As the derivative of the spiking activation function is zero everywhere (except at the threshold where it is ill-defined), a surrogate gradient must be used for backpropagation [35]. For the standard SpikeAct, a scaled version of the derivative of the $arctan$ function is used [35]. For SpikeAct$_{signed}$, we propose to use the sum of the surrogate of SpikeAct centered on the positive threshold and centered on the negative threshold, with the maximum being scaled to 1 (Fig. 2):

$$(SpikeAct_{signed})'(x) \approx \frac{1}{1+\frac{1}{1+\gamma*4*V_{th}^2}} \\ *(\frac{1}{1+\gamma*(x-V_{th})^2}+\frac{1}{1+\gamma*(x+V_{th})^2}) \quad (9)$$

with $\gamma = 10$ and $V_{th} = 1$. Moreover, we propose to add a reset gate (r) in the cell model of SpikGRU. Indeed, the original SpikGRU is defined with a single gate (the update gate z), while ANN GRUs (with two gates [6]) are used in ALR tasks [19, 43]. Indeed, GRU show good performance with a lower cost than models with more gates (such as LSTM). Therefore, SpikGRU2+ matches more closely the ANN GRU, while maintaining the essence of SpikGRU [11] (i.e. applying the spiking activation function only at the output of the neuron cell). SpikGRU2+ is illustrated in Fig. 3

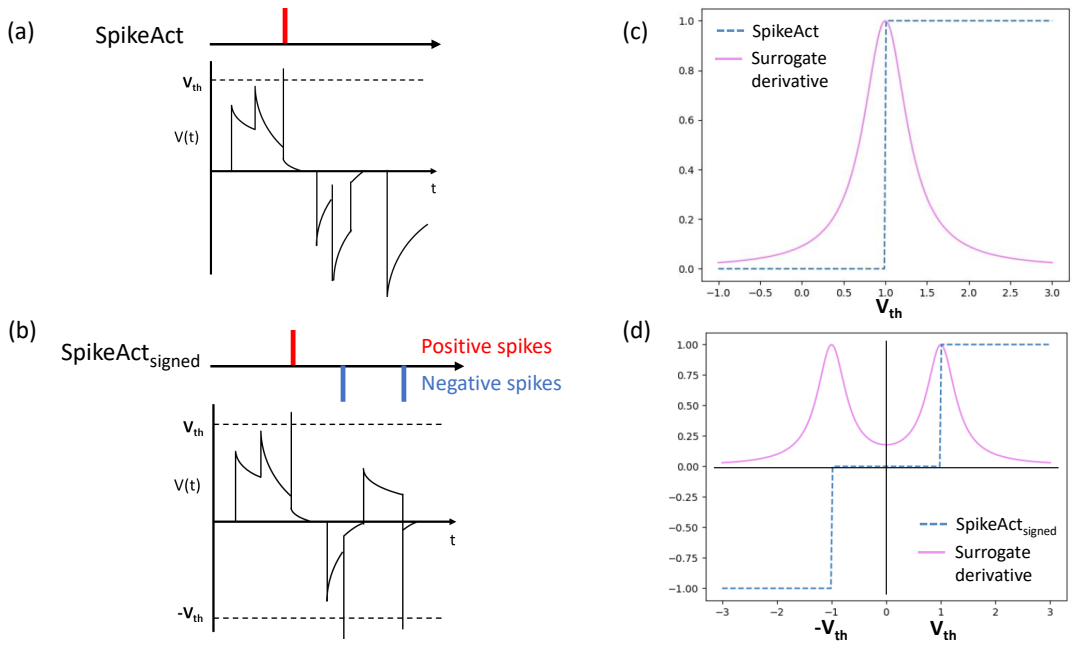

Figure 2. Membrane potential V(t) and output spikes of a SNN neuron, with SpikeAct (a) or SpikAct$_{signed}$ (b) activation functions (equations 1-4). SpikeAct (c) and SpikAct$_{signed}$ (d) activation functions and their surrogate derivatives used for gradient descent.

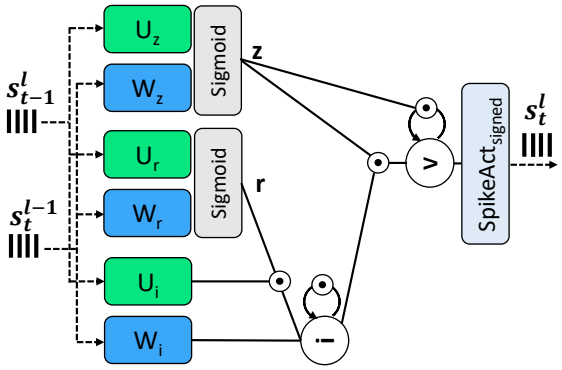

Figure 3. SpikGRU2+: a Signed Spiking Gated Recurrent Unit. The spiking synaptic operations are symbolized with dashed lines, and the full-precision vector operations are symbolized with solid lines (see equations 10-14). Biases are omitted for clarity.

and described as:

$$r_t^l = \sigma(W_r s_t^{l-1} + U_r s_{t-1}^l + b_r) \tag{10}$$

$$z_t^l = \sigma(W_z s_t^{l-1} + U_z s_{t-1}^l + b_z) \tag{11}$$

$$i_t^l = \alpha \odot i_{t-1}^l + W_i s_t^{l-1} + b_i + (U_i s_{t-1}^l + b_{ri}) \odot r_t^l \tag{12}$$

$$v_t^l = z_t^l \odot v_{t-1}^l + (1 - z_t^l) \odot i_t^l - v_{th} s_{t-1}^l \tag{13}$$

$$s_t^l = SpikeAct_{signed}(v_t^l) \tag{14}$$

3.2. Neural Network Topology

Following previous event-based lip-reading methods [3, 43], the neural network architecture is composed of a frontend based on ResNet-18 and a backend with a bi-directional GRU (Fig. 4). The first 2D convolutional layer of ResNet-18 is modified to be 3D with a kernel 5x7x7, and the max pooling layer is replaced by an average pooling. The output of ResNet frontend goes through a global average pooling before being input to a 3-layer bi-directional GRU with hidden size 1024. The output of the GRU is averaged in the temporal dimension and processed with a Fully Connected (FC) layer. ANN and SNN versions of the architecture are implemented. For the SNN version, the proposed SpikGRU2+ model is used for the backend. For the frontend, a spiking version of ResNet18 is implemented. The Spiking ResNet simply replaces the ANN Rectified Linear Unit (ReLU) activation functions with SpikeAct in the ResNet blocks, after the 2D convolutions and Batch Normalization (BN) layers (Fig. 5), as in [23, 49]. The Spiking ResNet layers use Parametric-LIF (PLIF) [18] neurons with learnable neuron leak β per channel (equation 1), while the average pooling layers use integrate-and-fire neurons (no leak). The standard spiking activation function (SpikeAct) is used in the Spiking ResNet frontend.

3.3. Dataset and Pre-processing

The DVS-Lip dataset [43] is composed of 100 classes of words from the vocabulary of the LRW dataset [7]. It con-

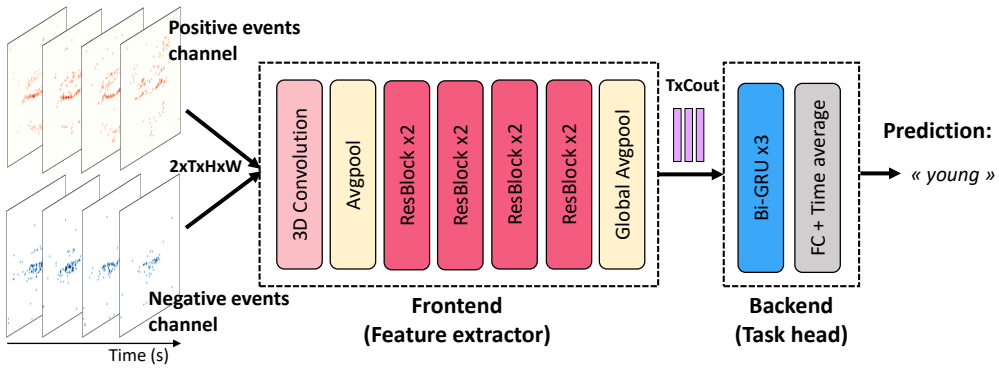

Figure 4. Model architecture. The frontend is a modified ResNet-18 and the backend is a 3-layer bi-directional GRU.

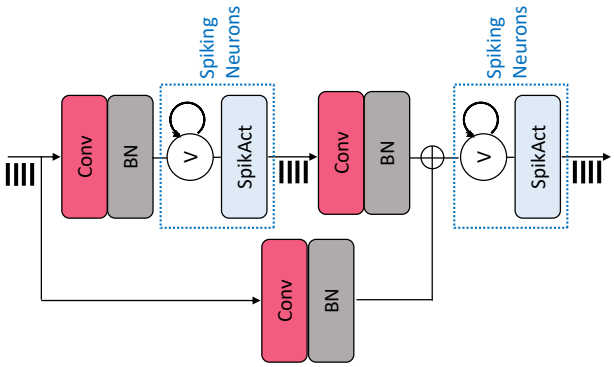

Figure 5. Spiking ResNet block with downsampling. Compared to the original ANN model, the ReLU activations are replaced with spiking (PLIF) neurons, composed of a stateful membrane potential V and a spiking activation function.

tains 19,871 samples recorded from 40 individuals, with 14,896 samples from 30 individuals used for training while the 4,975 samples from the remainder 10 individuals are used for evaluation as in [43]. The task is challenging because the individuals in the training and the evaluation set do not overlap, challenging the model to generalize to various speakers characteristics; and half of the words in the dataset correspond to the most frequently confused pairs of the LRW dataset (e.g. 'america' and 'american', etc.). Each sample lasts about 1 second, in which there are about 10^4 events generated with a spatial resolution of 128x128 pixels and a temporal resolution in the order of the microsecond.

SNNs directly process the raw input events asynchronously in neuromorphic hardware. However, in these simulations, the input events must be converted to frames for training with backpropagation through time. Temporal bins are accurately reproduced in timestep-synchronized neuromorphic hardware (such as Intel's Loihi [15]), allowing to combine neuron synchronization mechanisms and asynchronous event-based spike processing, fully grasping SNN benefits. T temporal bins per samples are used and events are associated to the closest temporal bin as in [3, 43]. Contrary to [3, 43], we did not weight the polarity of an event according to its distance to the closest temporal bin, which would prevent from using the raw events as input to the network in the neuromorphic implementation, and did not have a significant impact on the accuracy. Different channels are used for positive and negative polarity events. The samples are cut after 1.2s and 90 temporal bins are used, resulting in a timestep duration of 13ms (75Hz), and frames are center cropped to 88x88 in Height x Width.

3.4. Data Augmentation

Data augmentation is applied on the event frames (Fig. 6). The frames are center cropped to 96x96 then random cropped to 88x88, and horizontally flipped with 0.5 probability. This data augmentation, used in MSTP [43], is called **baseline**. Moreover, other **spatial augmentation** techniques are used: (1) a 2D spatial masking (cutout as in [25]) with 4 masks of maximal of maximum length 20; (2) zoom in and zoom out with a maximum zoom factor corresponding to 30% of the spatial size. Moreover, a **temporal augmentation** is proposed: temporal masks are applied to the frames using several masks with a given maximal length. This is equivalent to perform event dropping in time [20] with several intervals instead of a single one, which we found much more effective (see the ablation study). In these experiments, 6 temporal masks with a maximal length set to 18 frames yielded the best accuracy. In both temporal and spatial augmentations, the length of masks (and zoom factor) are randomly sampled between zero and the maximum length (or factor).

3.5. Training details

The neural networks are implemented with the Pytorch framework. Adam [22] optimizer is used with standard settings. The enhanced data augmentation increases the convergence time, and SNNs also converge slower than ANNs. SNNs are trained during 100 epochs (with batch size 32)

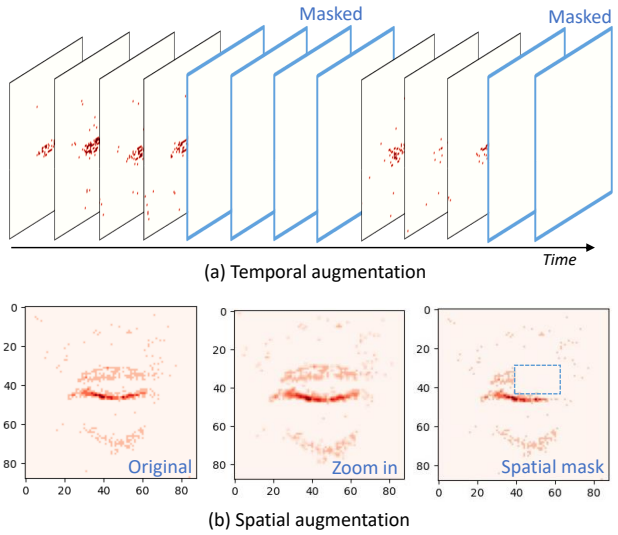

Figure 6. Spatial and temporal data augmentation. Blue annotations indicate the modifications compared to the original data.

with a fixed learning rate of 3e-4, and then during 100 epochs with a decaying learning rate (using a cosine annealing scheduler [27]). ANNs are directly trained with the decaying learning rate for 100 epochs. A warmup epoch (with a learning rate starting from 0 and linearly increasing up to initial learning rate) is used for all models. A weight decay of 1e-4 is applied to all parameters, except BN parameters. Dropout is used after the ResNet, the GRU, and on each hidden layer of the GRU, with probability 0.5, 0.5, 0.2 respectively. All models are trained with a NVIDIA A100-SXM4 GPU with 40GB memory. SNN training is longer than ANN training, due to the simulation of spiking neurons requiring sequential processing.

3.6. Spike Loss

In event-based neuromorphic implementations, the number of operations per second is proportional to the spike sparsity of the SNN [13]. Although SNN can naturally be sparse, their sparsity can be enhanced with training, by adding a spike loss to the accuracy loss. This loss can be computed directly with the number of spikes [12, 36]. However, in these experiments, the spikes are weighted by the number of synaptic connections they impact, as it represents with higher fidelity the resulting number of operations [13]:

$$Loss_{spike} = \lambda \frac{\sum_{l \in layers} (\frac{1}{2TN} \sum_t (s_t^{in^l})^2) * Syn^l}{\sum_{l \in layers} Syn^l} \quad (15)$$

$s_t^{in^l}$ are the input spikes of layer l at timestep $t \in T$, which corresponds to the output spikes of the N neurons of layer $l-1$ (respectively l) for feedforward (respectively recurrent) connections. Syn^l is the number of synapses of layer l and λ is a hyperparameter. The spikes are squared to ensure that the derivative is zero (i.e. the loss is not applied) when there is no spikes (i.e. $s_t^{in^l} = 0$) [36]. This loss is only used during additional fine-tuning of the best SNN model. Experiments are performed with different λ (from 1 to 100) to obtain models with different spike sparsity.

4. Results

4.1. Comparison with State-of-the-art

Results on the DVS-Lip dataset are compared with the state-of-the-art ANN and SNN models in Table 1. Notably, both our SNN and ANN outperform the previous state-of-the-art by 4% with a simpler model architecture (indeed, the MSTP frontend uses a second smaller ResNet-18 in parallel of the ResNet-18 and additional convolutions to merge the two branches [43]). For the ANN, this is explained by the better data augmentation technique, as will be shown in the ablation study. In addition, our SNN largely outperforms the state-of-the-art SNN (by 25%), due to both a better data augmentation and the proposed SpikGRU2+ backend, as will be shown in the Backend ablation study.

The fact that our SNN is as accurate as our ANN baseline despite the low-precision spiking activations can be explained by several reasons: (1) the enhanced SNN backend with signed spiking activations and full-precision neuronal states; (2) the event-based nature of data, which matches the SNN activation dynamics; (3) the use of a relatively large topology for the task. Indeed, [32] have demonstrated up to 69.4% accuracy with less than 1M parameters. While it has been shown that the accuracy of SNNs significantly increases with the width of layers (due to the low-resolution activations) [14], this width may not be as beneficial for the full-precision ANN.

4.2. Computational and Energy Efficiency

The computational and energy efficiency of the proposed SNN is compared with the ANN baseline. Fine-tuning the best SNN model with the spike loss, using different coefficient λ, allows to obtain SNNs with different accuracy and spike sparsity (Table 2). Notably, fine-tuning with a small λ and a smaller learning rate does not decrease much the number of spikes per synapse (equivalent to number of operations per synapse, OPs/syn, in a SNN) but leads to a slight increase in accuracy (75.3%). Conversely, increasing λ and the fine-tuning learning rate significantly reduces the number of spikes per synapse (and so the number of operations), but degrades the accuracy. Note that the frontend causes the majority of the operations (26.0 GOPs/s vs. 3.6 GOPs/s in the backend, for the ANN), although the backend has a higher number of parameters (47M vs. 11M). However, the spike sparsity is higher in the frontend (0.036 to 0.088 OPs/syn) than in the backend (0.099 to 0.382 OPs/syn). In

Paper	Type	Frontend	Backend	Nb. params (M)	Acc. (%)
MSTP [43]	ANN	MSTP ResNet-18	BiGRU	60.3	72.1
MSTP [43]	ANN	ResNet-18	BiGRU	58.6	70.7
Ours	ANN	ResNet-18	BiGRU	58.6	**75.1**
Spiking MSTP [3]	SNN	SEW-ResNet18	FC (stateful)	11.3	60.2
Spiking MSTP [3]	SNN	SEW-ResNet18	Spiking BiGRU (adapted from [28])	58.6	46.3
Ours	SNN	Spiking ResNet-18	Spiking BiGRU (**SpikGRU2+**)	58.6	**75.3**

Table 1. Comparison with the state-of-the-art ANN and SNN on the DVS-Lip [43] dataset.

	Acc. (%)	OPs/syn	GOPs/s	Energy red.[1]
ANN	75.1	1	29.5	1x
SNN	75.1	0.128	3.8	1.2 - 11x
With spike loss fine-tuning				
SNN	75.3	0.124	3.7	1.3 - 11x
SNN	74.5	0.099	3.0	1.6 - 14x
SNN	73.8	0.069	2.1	2.3 - 20x
SNN	73.3	0.044	1.3	3.6 - 31x

Table 2. Efficiency of SNNs trained with spike loss vs. ANN, depending on the spike sparsity (i.e. number of operations per synapse). [1]Energy reduction estimated using models of the energy consumption of memory accesses and operations in a SNN vs. an ANN accelerator, depending on SNN spike sparsity [10, 13].

overall, the SNNs reduce the number of operations from 8x to 22x compared to the ANN, with +0.2% to -1.8% accuracy difference. Note that the spike loss is defined to minimize the spike rate (average number of spikes per timestep) while the timesteps rate is assumed fixed. In future work, the timesteps rate could be jointly considered with the spike rate to better optimize the number of spikes per second (product of timesteps per second and spikes per timestep).

Furthermore, the energy efficiency of SNNs in an event-based neuromorphic implementation compared with ANN accelerators is estimated, accounting not only for the number of operations, but also the associated memory accesses, which are responsible for most of the energy consumption [13]. In order to be agnostic to a specific accelerator, the models of the energy efficiency of SNNs relative to ANNs from [10, 13] are used. Indeed, theoretical lower bound (considering a non-optimized ANN accelerator) and upper bound (considering a maximally-optimized ANN accelerator) are defined depending on the SNN spike sparsity, accounting for the dynamic energy consumption related to memory accesses and operations in standard digital accelerators. According to this model, the event-based implementations of the proposed SNNs are expected to be 1.3x to 31x more energy-efficient than the ANN implementation, depending on the SNN accuracy and the optimization of the ANN implementation.

Baseline ([43])	Spatial aug.	Temporal aug.	Acc. (%)
✓			70.1
✓	✓		71.5
✓		✓	73.1
✓	✓	✓	75.1

Table 3. Data augmentation impact on test accuracy. Spatial augmentation consists in spatial masking, as well as zoom in and zoom out, and temporal augmentation consists in time masking.

Nb. masks	4	6	8	60	3	1
Max size	18	18	18	1	30	60
Acc. (%)	73.2	75.1	74.4	70.3	73.7	68.9

Table 4. Effect of the hyperparameters (number of masks and maximum length of a mask) of the temporal masking data augmentation on the test accuracy.

4.3. Ablation Study

Data augmentation

The effectiveness of the proposed data augmentation is studied on the ANN model and compared to the data augmentation used in previous state-of-the-art [3, 43] in Table 3. Both spatial and temporal augmentation significantly improves the accuracy. The effect of the hyperparameters of the temporal masking (number of masks and maximum length of a mask) is studied in Table 4. It is observed that random masking of frames does not improve the accuracy, but rather that the masks must have a sufficient length. We suggest that it helps reducing the important overfitting of the GRU backend, by forcing the GRU to preserves information for a longer time (while a sequence of frames are masked). In addition, several masks (allowing to mask several portions of the input) yield higher accuracy compared to a single mask as proposed in [20].

Backend	SpikeAct$_{signed}$	Nb. of gates	Acc. (%)
FC (as [3])	n.a.	n.a.	68.1
SpikGRU		1	68.7
SpikGRU2		2	70.6
SpikGRU+	✓	1	74.2
SpikGRU2+	✓	2	75.1
GRU (ANN)	n.a.	2	73.2

Table 5. Impact of the backend on the test accuracy. The frontend is a Spiking ResNet-18 (PLIF neurons with SpikeAct). ANN and SNN GRU backends are bi-directional.

Backend

Different backends with the same frontend (Spiking ResNet-18) are compared in Table 5. The backend of Spiking MSTP [3], a FC layer with stateful synapses, is implemented as baseline. It yields higher accuracy (+11%) than in [3], which can be due to (1) the better data augmentation technique; (2) the higher frame rate (T=90 vs. T=30), which has been shown to improve accuracy [43]. The original SpikGRU [11] is compared against the proposed SpikGRU2+, showing the large accuracy gap between the two models (68.7% vs. 75.1%). The benefits of the SpikeAct$_{signed}$ and the additional gate are shown separately. SpikeAct$_{signed}$ is shown to have the highest impact, although the second gate also significantly improves the accuracy. An ANN backend (as in the full ANN) is also implemented, which surprisingly yielded lower accuracy. This could be explained by the higher overfitting of the ANN, or the non-optimality of the mixed ANN-SNN training.

Spiking activation function

The impact of using Signed SNNs (with SpikeAct$_{signed}$ activation function) instead of traditional SNN (with SpikeAct activation function), in the frontend and/or the backend, on both accuracy and spike rate, is evaluated in Table 6 and Fig. 7. The fully Signed SNN (in both frontend and backend) achieves higher accuracy than the standard SNN, but lower accuracy than the hybrid SNN (standard in the frontend and signed in the backend). We hypothesize that symmetric activation function (as SpikeAct$_{signed}$) may promote stability of the GRU (as $tanh$ in ANN RNNs). On the contrary, in CNN layers, positive-only spiking activations are more effective (as ReLU in ANN CNNs), and using SpikeAct$_{signed}$ only increases overfitting. In addition, as shown in Fig. 7, SpikeAct$_{signed}$ reduces the spike sparsity (note that the spike loss fine-tuning is not used in this ablation). This could be explained by the non-spiking region of the membrane potential being reduced to $]-v_{th}, +v_{th}[$ with SpikeAct$_{signed}$, instead of $]-\infty, +v_{th}[$ (equations 3-4). Nevertheless, as shown in the efficiency evaluation, the impact of the backend in the total number of operations is small compared to the frontend, and spike sparsity can be enhanced by optimization. Hence, the Hybrid SNN benefits from the high accuracy of the signed backend while maintaining a high efficiency.

	Frontend	Backend	Acc. (%)
SNN	SpikeAct	SpikeAct	70.6
Signed SNN	SpikeAct$_{signed}$	SpikeAct$_{signed}$	72.3
Hybrid SNN	SpikeAct	SpikeAct$_{signed}$	75.1

Table 6. SNNs with different spiking activation function (SpikeAct or SpikeAct$_{signed}$) for the frontend and backend.

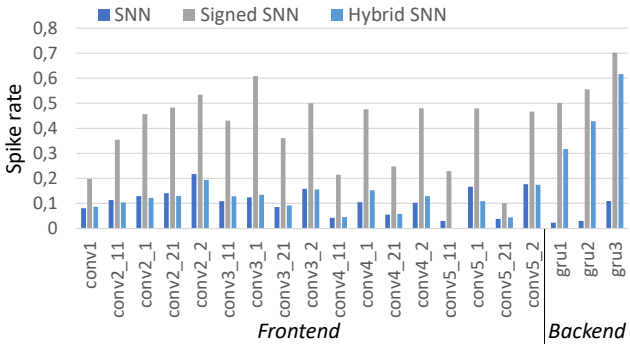

Figure 7. Spike rate (average number of spikes produced per neuron per timestep) of the three SNN versions from Table 6.

5. Conclusion

Event-based cameras and Spiking Neural Networks in an end-to-end neuromorphic implementation can allow efficient automatic lip-reading on portable devices. In this work, a spiking model based on a Signed Spiking Gated Recurrent Unit (SpikGRU2+) task head achieving state-of-the-art accuracy on the challenging DVS-Lip dataset is proposed. Moreover, effective spatial and temporal data augmentation techniques for event-based gesture recognition are provided. With the sparsity fine-tuning, the SNN reduces the number of operations from 8x to 22x compared to the ANN, with +0.2% to -1.8% accuracy difference. The event-based neuromorphic implementation is expected to be 1.2x to 31x more energy-efficient than a dedicated ANN accelerator. As future work, mapping the SNN onto an event-based neuromorphic accelerator would showcase the energy consumption savings. Smaller and quantized models will also be considered to further improve efficiency. The applicability of the method to other event-based recognition tasks should be investigated. This work opens up new perspectives for the use of SNNs for accurate and low-power neuromorphic gesture recognition.

References

[1] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, Jeff Kusnitz, Michael Debole, Steve Esser, Tobi Delbruck, Myron Flickner, and Dharmendra Modha. A low power, fully event-based gesture recognition system. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7388–7397, 2017. 2, 3

[2] Sami Barchid, José Mennesson, and Chaabane Djéraba. Exploring joint embedding architectures and data augmentations for self-supervised representation learning in event-based vision. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3903–3912, 2023. 3

[3] Hugo Bulzomi, Marcel Schweiker, Amélie Gruel, and Jean Martinet. End-to-end neuromorphic lip reading. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4101–4108, 2023. 2, 3, 4, 5, 7, 8

[4] Kenneth Chaney, Artemis Panagopoulou, Chankyu Lee, Kaushik Roy, and Kostas Daniilidis. Self-supervised optical flow with spiking neural networks and event based cameras. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5892–5899, 2021. 2

[5] Xuena Chen, Li Su, Jinxiu Zhao, Keni Qiu, Na Jiang, and Guang Zhai. Sign language gesture recognition and classification based on event camera with spiking neural networks. *Electronics (Switzerland)*, 12(4), 2023. 2, 3

[6] 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)*, pages 1724–1734, 2014. 3

[7] J.S. Chung and A. Zisserman. Lip reading in the wild. In *Asian Conference on Computer Vision*, 2016. 3, 4

[8] Javier Cuadrado, Ulysse Rançon, Benoit R. Cottereau, Francisco Barranco, and Timothée Masquelier. Optical flow estimation from event-based cameras and spiking neural networks. *Frontiers in Neuroscience*, 17, 2023. 2

[9] Thomas Dalgaty, Thomas Mesquida, Damien Joubert, Amos Sironi, Cyrille Soubeyrat, Pascal Vivet, and Christoph Posch. The cnn vs. snn event-camera dichotomy and perspectives for event-graph neural networks. In *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6, 2023. 2

[10] Manon Dampfhoffer. *Models and algorithms for implementing energy-efficient spiking neural networks on neuromorphic hardware at the edge*. Theses, Université Grenoble Alpes, 2023. 7

[11] 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*, pages 359–370, 2022. 2, 3, 8

[12] Manon Dampfhoffer, Thomas Mesquida, Emmanuel Hardy, Alexandre Valentian, and Lorena Anghel. Leveraging sparsity with spiking recurrent neural networks for energy-efficient keyword spotting. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023. 6

[13] 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*, 7(3):731–741, 2023. 6, 7

[14] Manon Dampfhoffer, Thomas Mesquida, Alexandre Valentian, and Lorena Anghel. Backpropagation-based learning techniques for deep spiking neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–16, 2023. 2, 6

[15] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, Yuyun Liao, Chit-Kwan Lin, Andrew Lines, Ruokun Liu, Deepak Mathaikutty, Steven McCoy, Arnab Paul, Jonathan Tse, Guruguhanathan Venkataramanan, Yi-Hsin Weng, Andreas Wild, Yoonseok Yang, and Hong Wang. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018. 2, 5

[16] Haowen Fang, Amar Shrestha, Ziyi Zhao, and Qinru Qiu. Exploiting neuron and synapse filter dynamics in spatial temporal learning of deep spiking neural network. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2799–2806, 2020. 2, 3

[17] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep Residual Learning in Spiking Neural Networks. In *Advances in Neural Information Processing Systems*, pages 21056–21069, 2021. 3

[18] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2641–2651, 2021. 2, 4

[19] Dalu Feng, Shuang Yang, Shiguang Shan, and Xilin Chen. Learn an effective lip reading model without pains. *arXiv:2011.07557*, 2020. 3

[20] Fuqiang Gu, Weicong Sng, Xuke Hu, and Fangwen Yu. Eventdrop: Data augmentation for event-based learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 700–707, 2021. 3, 5, 7

[21] Jesse Hagenaars, Federico Paredes-Vallés, and Guido De Croon. Self-supervised learning of event-based optical flow with spiking neural networks. *Advances in Neural Information Processing Systems*, 34:7167–7179, 2021. 2

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 5

[23] Chankyu Lee, Syed Shakib Sarwar, Priyadarshini Panda, Gopalakrishnan Srinivasan, and Kaushik Roy. Enabling

Spike-Based Backpropagation for Training Deep Neural Network Architectures. *Frontiers in Neuroscience*, 14:119, 2020. 4

[24] Chen Li, Lei Ma, and Steve Furber. Quantization Framework for Fast Spiking Neural Networks. *Frontiers in Neuroscience*, 16, 2022. 2

[25] Yuhang Li, Youngeun Kim, Hyoungseob Park, Tamar Geller, and Priyadarshini Panda. Neuromorphic data augmentation for training spiking neural networks. In *Computer Vision – ECCV 2022*, pages 631–649, 2022. 3, 5

[26] Patrick Lichtensteiner, Christoph Posch, and T. Delbruck. A 128x128 120db 15µs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 2:566–576, 2008. 1, 3

[27] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017*, 2017. 6

[28] Ali Lotfi Rezaabad and Sriram Vishwanath. Long short-term memory spiking networks and their applications. In *International Conference on Neuromorphic Systems 2020*, pages 1–9. ACM, 2020. 2, 3, 7

[29] Pingchuan Ma, Stavros Petridis, and Maja Pantic. Visual speech recognition for multiple languages in the wild. *Nature Machine Intelligence*, 4:930–939, 2022. 1

[30] Carver Mead. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):1629–1636, 1990. 1

[31] Paul A. Merolla, John V. Arthur, Rodrigo Alvarez-Icaza, Andrew S. Cassidy, Jun Sawada, Filipp Akopyan, Bryan L. Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, Bernard Brezzo, Ivan Vo, Steven K. Esser, Rathinakumar Appuswamy, Brian Taba, Arnon Amir, Myron D. Flickner, William P. Risk, Rajit Manohar, and Dharmendra S. Modha. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197): 668–673, 2014. 2

[32] Thomas Mesquida, Manon Dampfhoffer, Thomas Dalgaty, Pascal Vivet, Amos Sironi, and Christoph Posch. G2N2: Lightweight Event Stream Classification with GRU Graph Neural Networks. In *34th British Machine Vision Conference 2023, BMVC*, 2023. 3, 6

[33] Daniel Michelsanti, Zheng-Hua Tan, Shi-Xiong Zhang, Yong Xu, Meng Yu, Dong Yu, and Jesper Jensen. An overview of deep-learning-based audio-visual speech enhancement and separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1368–1396, 2021. 1

[34] Youssef Mroueh, Etienne Marcheret, and Vaibhava Goel. Deep multimodal learning for audio-visual speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2130–2134, 2015. 1

[35] 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*, 36:51–63, 2019. 3

[36] 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*, pages pp. 97–103, 2021. 6

[37] Michael Pfeiffer and Thomas Pfeil. Deep learning with spiking neurons: Opportunities and challenges. *Frontiers in Neuroscience*, 12, 2018. 2

[38] Ali Samadzadeh, Fatemeh Sadat Tabatabaei Far, Ali Javadi, Ahmad Nickabadi, and Morteza Haghir Chehreghani. Convolutional spiking neural networks for spatio-temporal feature extraction. *Neural Processing Letters*, 55(6):6979 – 6995, 2023. 2, 3

[39] Brendan Shillingford, Yannis Assael, Matthew W. Hoffman, Thomas Paine, Cían Hughes, Utsav Prabhu, Hank Liao, Hasim Sak, Kanishka Rao, Lorrayne Bennett, Marie Mulville, Misha Denil, Ben Coppin, Ben Laurie, Andrew Senior, and Nando de Freitas. Large-Scale Visual Speech Recognition. In *Proc. Interspeech 2019*, pages 4135–4139, 2019. 3

[40] Mitchell S. Sommers, Nancy Tye-Murray, and Brent Spehar. Auditory-visual integration and lipreading abilities of older adults with normal and impaired hearing. *The Journal of the Acoustical Society of America*, 120:3347–3347, 2006. 1

[41] Themos Stafylakis and Georgios Tzimiropoulos. Combining Residual Networks with LSTMs for Lipreading. In *Proc. Interspeech 2017*, pages 3652–3656, 2017. 3

[42] Pengfei Sun, Longwei Zhu, and Dick Botteldooren. Axonal delay as a short-term memory for feed forward deep spiking neural networks. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8932–8936, 2022. 2

[43] Ganchao Tan, Yang Wang, Han Han, Yang Cao, Feng Wu, and Zheng-Jun Zha. Multi-grained spatio-temporal features perceived network for event-based lip-reading. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20062–20071, 2022. 2, 3, 4, 5, 6, 7, 8

[44] Yuchen Wang, Malu Zhang, Yi Chen, and Hong Qu. Signed neuron with memory: Towards simple, accurate and high-efficient ann-snn conversion. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 2501–2508, 2022. Main Track. 2

[45] Shuang Yang, Yuanhang Zhang, Dalu Feng, Mingmin Yang, Chenhao Wang, Jingyun Xiao, Keyu Long, Shiguang Shan, and Xilin Chen. Lrw-1000: A naturally-distributed large-scale benchmark for lip reading in the wild. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, page 1–8. IEEE Press, 2019. 3

[46] Bojian Yin, Federico Corradi, and Sander M. Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10):905–913, 2021. 2

[47] Sangmin Yoo, Eric Yeu-Jer Lee, Ziyu Wang, Xinxin Wang, and Wei D. Lu. Rn-net: Reservoir nodes-enabled neuromorphic vision sensing network. *arXiv:2303.10770*, 2023. 3

[48] Malu Zhang, Jibin Wu, Ammar Belatreche, Zihan Pan, Xiurui Xie, Yansong Chua, Guoqi Li, Hong Qu, and Haizhou Li.

Supervised learning in spiking neural networks with synaptic delay-weight plasticity. *Neurocomputing*, 409:103–118, 2020. 2

[49] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11062–11070, 2021. 4