Attentive Decision-making and Dynamic Resetting of Continual Running SRNNs for End-to-End Streaming Keyword Spotting

Bojian Yin
Bojian.Yin@cwi.nl
CWI
Amsterdam, The Netherlands

Qinghai Guo
guoqinghai@huawei.com
Huawei
Shenzhen, China

Henk Corporaal
h.corporaal@tue.nl
TU/e
Eindhoven, The Netherlands

Federico Corradi
f.corradi@tue.nl
TU/e
Eindhoven, The Netherlands

Sander M. Bohté
S.M.Bohte@cwi.nl
CWI
Amsterdam, The Netherlands

ABSTRACT

Efficient end-to-end processing of continuous and streaming signals is one of the key challenges for Artificial Intelligence (AI) in particular for Edge applications that are energy-constrained. Spiking neural networks are explored to achieve efficient edge AI, employing low-latency, sparse processing, and small network size resulting in low-energy operation. Spiking Recurrent Neural Networks (SRNNs) achieve good performance on sample data at excellent network size and energy. When applied to continual streaming data, like a series of concatenated keyword samples, SRNNs, like traditional RNNs, recognize successive information increasingly poorly as the network dynamics become saturated. SRNNs process concatenated streams of data in three steps: i) Relevant signals have to be localized. ii) Evidence then needs to be integrated to classify the signal, and finally, iii) the neural dynamics must be combined with network state resetting events to remedy network saturation.

Here we show how a streaming form of attention can aid SRNNs in localizing events in a continuous stream of signals, where a brain-inspired decision-making circuit then integrates evidence to determine the correct classification. This decision then leads to a delayed network reset, remedying network state saturation. We demonstrate the effectiveness of this approach on streams of concatenated keywords, reporting high accuracy combined with low average network activity as the attention signal effectively gates network activity in the absence of signals. We also show that the dynamic normalization effected by the attention mechanism enables a degree of environmental transfer learning, where the same keywords obtained in different circumstances are still correctly classified. The principles presented here also carry over to similar applications of classical RNNs and thus may be of general interest for continual running applications.

ICONS 2022, July 27–29, 2022, Knoxville, TN, USA

ACM ISBN 978-1-4503-9789-6/22/07.
https://doi.org/10.1145/3546790.3546795

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

datasets, neural networks, gaze detection, text tagging

ACM Reference Format:
Bojian Yin, Qinghai Guo, Henk Corporaal, Federico Corradi, and Sander M. Bohté. 2022. Attentive Decision-making and Dynamic Resetting of Continual Running SRNNs for End-to-End Streaming Keyword Spotting. In *International Conference on Neuromorphic Systems 2022 (ICONS 2022), July 27–29, 2022, Knoxville, TN, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3546790.3546795

1 INTRODUCTION

Many observational tasks are inherently of an intermittent and continuous nature: while one has to continuously observe surroundings for dangers, the proverbial tiger is fortunately present most sparingly. In a more applied context, keyword spotting requires a similar continuous, or *streaming*, environmental monitoring with relevant stimuli appearing relatively rarely. In each case, a proper balance has to be found between the false alarm rate (seeing a tiger where there is none) and the false reject rate (overlooking the tiger).

Continuous online processing of streaming information is a particular challenge in energy-constrained situations such as applications running on battery-operated devices. Event-based neural networks like spiking neural network are explored as a means to achieve both low-latency and sparse neural processing, and Spiking Recurrent Neural Networks (SRNNs) in particular achieve good performance on sample data at excellent network size and energy. When continually applied on streaming data however, for example a series of concatenated keyword samples with or without extended pauses, SRNNs, like traditional RNNs, recognize successive information increasingly poorly as the network dynamics become saturated [2]. For RNNs, including modern transformer-based variants like the Conformer [5], solutions have been sought in periodically resetting the internal state of the network, where resetting is typically done using empirical measures tuned for the task at hand [2].

Here, we take inspiration from biology to dynamically reset compact SRNNs to process concatenated continuous streams in continually. For this, we introduce an efficient form of self-attention to localize relevant signals, which also gates information to be

integrated into the decision-making circuit to obtain a classification of the detected event. The actual classification is then used as a trigger for resetting the SRNN network state.

We show that compact spiking recurrent neural networks trained on single samples integrated into such extended circuitry can then successfully classify sequences of concatenated keywords. Moreover, they can do this without signal buffers or additional post-processing, demonstrating an efficient and compact end-to-end event-based solution. We also show that the dynamic normalization effected by our attention mechanism enables a degree of environmental transfer learning, where the same keywords obtained in different circumstances are still correctly classified.

2 BACKGROUND

Current approaches to continuous and streaming keyword spotting include three independent steps [18]. First, a stream is typically chunked into segments, for example, using Voice-Activation-Detection algorithms [2]. Second, each segment is processed using a set of overlapping fixed windows on the signal into a set of feature maps. Third, the concatenated features maps are processed to determine a classification label. Single windows can be processed into feature maps with learned approaches, such as Convolutional Neural Networks (CNNs), trained on single labeled samples. CNN's outputs are then converted into a sequence of labels using, for example, recurrent neural networks trained on the Connectionist Temporal Classification loss (CTC) [3]. These RNNs can also be replaced by modern transformers [5, 14] which improve performance but are still applied to segmented utterances. In each of these examples, a post-processing stage transduces observed signal sequences into a labeled sequence.

In [15], a spiking neural network (SNN) version of a CNN is applied to single keyword samples, demonstrating competitive performance with the equivalent non-spiking CNN. The WaveSense model [12] is derived from the classical WaveNet network [9] and is shown to effectively process single samples from various benchmarks up to 5s in length.

Still, these approaches rely on pre-processing to obtain segments and buffering to map sequences into labels. For energy-constrained continuous-monitoring applications, there is a need for continual running end-to-end SNN solutions that minimize pre- and post-processing and have minimal memory and processing requirements.

3 ATTENTIVE SPIKING RECURRENT NEURAL NETWORKS

We are specifically interested in continual online keyword recognition and localization in recurrent spiking neural networks, where the network omits buffering and only has access to the current information. To achieve this, we turn to a form of "attention" to help guide the recurrent spiking neural network in localizing and classifying utterances compatible with continual running.

Attention has been at the center of current Transformer models, where initially attention was introduced to learn long-range dependencies on image classification and Natural Language Processing (NLP) tasks [10]. In the context of limited or no buffering, attention in recurrent spiking neural networks only allows forms of local and causal self-attention without relying on long-term temporal dependencies. We observe that a straightforward measure of current signal-variability (Temporal Intensity) resembles a temporally local attention-like signal with the potential for localizing speech patterns in a sequence.

We define a specific version of Temporal Intensity based on the Mel-frequency spectrum representation of the signal, which is also the input to the network. We define a temporal average over a single time-step as $\mu_t = |x_t + x_{t-1}|/2$ and associated signal variability as $\sigma_t = |x_t - x_{t-1}|$ based on current input x_t. The real-time Temporal Intensity $tvar_t$ is then derived from μ_t and σ_t and rescaled to [0,1] by the *tanh* function:

$$tvar_t = \tanh(\eta \sigma_t \mu_t), \tag{1}$$

where η is a hyperparameter we emperically set to $\eta = 4$ in all experiments.

We further define a smoothed Temporal Intensity $tvar^s_t$ as $tvar^s_t = tvar^s_{t-1} + (1-\phi)(tvar_t - tvar^s_{t-1})$ where $\phi = \exp(-1/\tau_{tvar})$ determines the smoothness. This smoothed Temporal Intensity is used to facilitate the process of evidence accumulation along speech as the the curve of $tvar_t$ tends to be discontinuous (illustrated in Fig 1). In contrast to advanced attention models, our $tvar^s_t$ directly localizes speech patterns in ongoing speech sequences, is parameter free, and can be computed in an online manner.

The $tvar_t$ and $tvar^s_t$ measures are illustrated on several keywords speech audio samples and corresponding MFCC representation in Fig. 1: in the samples, we see that both measures map closely to the envelope of the signal.

SRNNs. To implement continual running spike-based RNNs, we use Adaptive Spiking Recurrent Neural Networks, SRNNs, as developed in [16], comprised of adaptive spiking neurons[1, 16]. Here, the SRNNs are comprised of an input layer converting the input spectrum into spikes. This input layer is densely connected to a single recurrent layer, where the $tvar^s_t$ is also added as an input to the recurrent layer. The final layer is comprised of leaky integrators to generate the prediction probabilities, p^i_t for the ith class at a timestep t. The structure is illustrated in Fig 1, described as a structure of 512D-(512+1)R-(12/36+1)I, where the number of output neurons (12 or 36) is task-dependent, and D denotes a dense layer, R the recurrent layer, and I the layer of integrators. The network omits any bias units as they proved detrimental for continual running. As illustrated in Fig 1, the SRNN reads the spectrum row by row at each time step, where we call each row a frame; the SRNN thus makes an online prediction at each time step.

We train the parameters of the SRNN using BPTT[16], with some modifications. In the continual running model, the SRNN needs to extract a class-probability label at every timestep. This means that also when learning, the SRNN needs to assign a label to each timestep. For pre-segmented samples however, in many cases only the label for the whole segment is given, and while the actual signal is somewhat centered, it is often flanked by silence or noisy frames. When trained on such pre-segmented samples, the ASRNN ideally only learns from the actual signal and not from the silent or noisy flanks. To achieve this, we introduce an instantaneous Temporal Intensity -gated loss-function between prediction $\hat{y_t}$ and target y for the labeled sample:

$$l_t = loss(y, \hat{y_t}) * tvar^s_t. \tag{2}$$

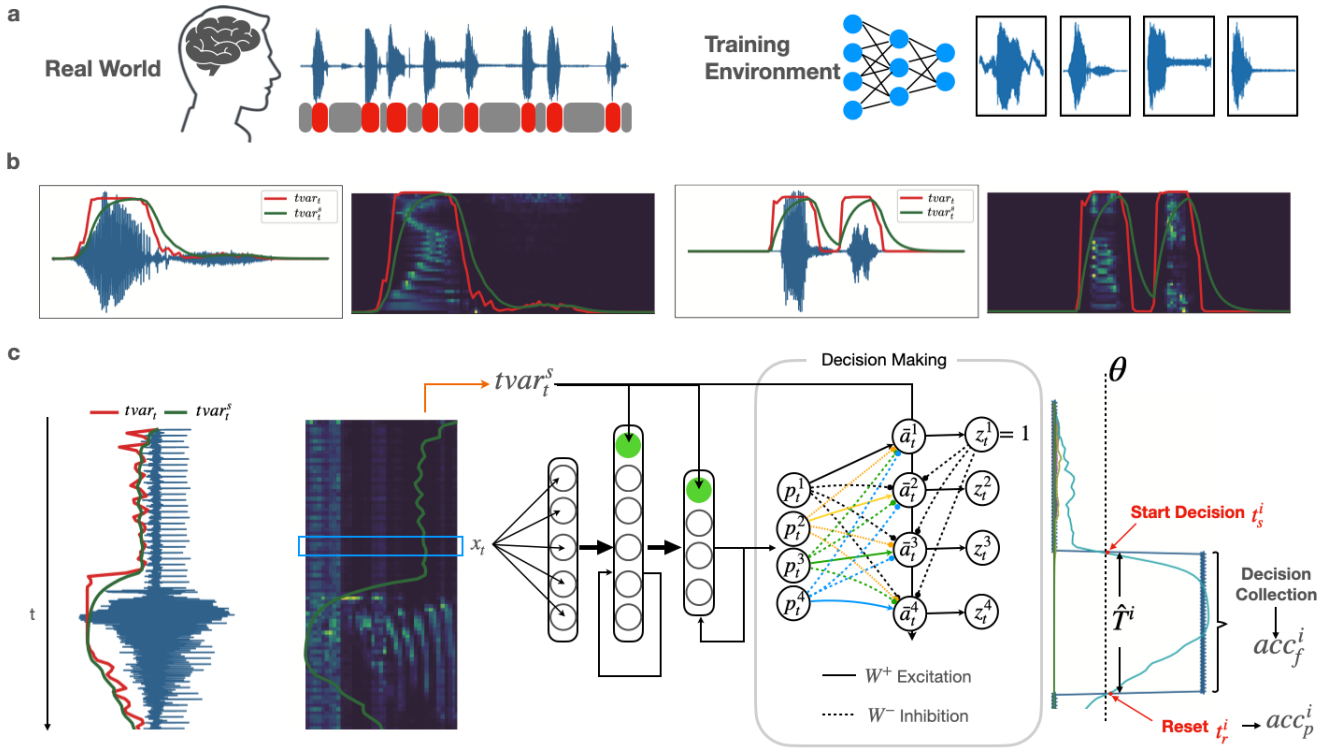

Figure 1: a) Continual running as decision-making vs single sample training; note the pauses in the speech signal where no utterance is present. b) Example of speech audio data and corresponding MFCC figure. The red and green curves correspond to the Temporal Intensity and smoothed Temporal Intensity measures. c) End-to-end decision-making procedure. The Temporal Intensity measures and MFCC spectrum are continually computed in an online manner. For each frame t, the smoothed Temporal Intensity and MFCC spectrum are fed into the SRNN, resulting in class probability outputs p_j^t. These outputs are integrated in the decision-making action value nodes a_j^t and associated gated values $\bar{a}_j^t$. Once a threshold θ is reached, the most activated action is selected and the other action values are suppressed. Once the action value for the selected action falls below a set threshold again, at the end of the utterance typically, the label is assigned and the network state is reset. The decision duration $\hat{T}^i = t_s^i, t_{s+1}, ..., t_r^i$ represents the period between the starting time of decision collection t_s^i and the reset timestep t_r^i. acc_f^i is the framewise accuracy and acc_p^i is the prediction accuracy for the sample i

As the Temporal Intensity calculates an envelope of the signal (i.e., Fig. 1b), this loss helps the network to learn primarily from actual signal data.

4 STREAMING DECISION MAKING

When a network continually generates class predictions for every frame, the challenge is to concatenate this sequence of class predictions into a sequence of predicted labels. Complex methods like the CTC exploit interdependence between frames or segments combined with implicit sequence modeling to determine the most likely sequence interpretation. However, in an online setting of concatenated keywords and silence/noise parts, labels are independent and temporally sparse, and the task is more closely related to sequential decision-making. We take inspiration from neural models of decision-making [4, 17], and introduce a decision-making circuit with dynamic resetting modeled after the Basal Ganglia brain structure, which is specifically involved in decision-making and context-dependent gating.

The decision-making circuit is shown in Fig. 1c, in the grey box: it accomplishes action selection by integrating class-probability inputs p_t^i for class i, where actions correspond to labels. An action is selected when a pre-defined threshold θ is reached, and results in the temporary inhibition of other actions. Resetting of the network is triggered when the integrated evidence falls below the threshold again while the Temporal Intensity signal is also rapidly declining at the same time (the effect of this latter condition is that in Fig. 2, for dynamic resetting, the third sample is correctly classified even though initially the wrong action/label is selected).

Action selection. The activity of the action selection system is modeled as a leaky integrator where a leak time-constant τ_ρ is associated with the typical duration of each action [17]. In the

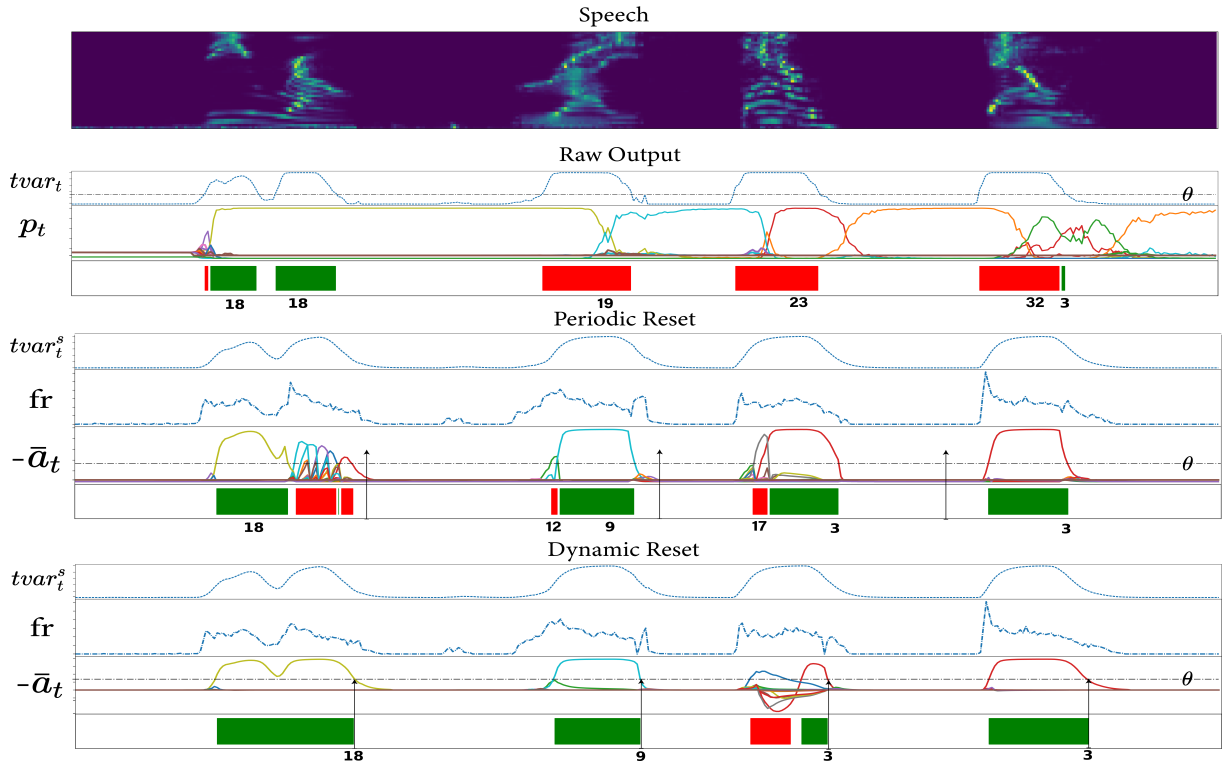

Figure 2: An example of the different decision-making process on concatenate speech audio sequence. Top: MFCC spectrum of four concatenated speech utterances. Next is plotted the Temporal Intensity as directly calculated and framewise classification probabilities p_t and resulting classifications (green: correct label, red: incorrect label). The plot below, 'Periodic Reset', demonstrates the effect of smoothing the Temporal Intensity ($tvar_t^s$), the resulting firing rate (fr) and gated action value ($\bar{a}_t$), and resulting framewise classifications given fixed periodic resets. The 'Dynamic Reset' plot illustrates what happens when resets instead are triggered by the decision-making circuit, and additionally winning actions inhibit other actions. Vertical black arrows denote the time of resets.

circuit, the action value a_t^i of class i at time t is computed as:

$$\begin{aligned} a_{t+1}^i &= a_t^i + (1-\rho)(u_t^i - a_t^i) \\ \text{where} \quad u_t^i &= -w_z^- I_t^i + w_z^+ \sum_{j \neq i}^{n} I_t^j, \end{aligned} \tag{3}$$

and where the input $I_t^i = p_t^i$ and $\rho = \exp(-dt/\tau_\rho)$. As in [4, 17] the balance between disinhibition and inhibition is chosen as $w_z^+/w_z^- = 1/n$, where n is the number of classes and $\tau_\rho = 20$, chosen to match the average speech length.

To pick up the prediction of the network only on the data, we use a gated action value as a conditional prediction probability or confidence to determine when speech is present. The gated action value is calculated as $\bar{a}_t^i = \tanh(tvar_t^s a_t^i)$ for class i. The frame-wise class label for each timestep is derived from the gated action with minimal value (maximal disinhibited) as well as

$$z_t^k = \begin{cases} 1, & \text{if } k = \underset{i \in 1,2,\ldots n}{\arg\min}\,(\bar{a}_t^i) \wedge \underset{i \in 1,2,\ldots n}{\min}\,(\bar{a}_t^i) < -\theta. \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

where we use a default value $\theta = 0.3$. Note that the same measure can be used as an indicator for speech/no-speech at time t.

Action Inhibition. Once an action (class label) is selected, all other classes are inhibited (where $z_t^j = 0, j \neq i$) when speech is **first** detected at time t. Inhibition is implemented by providing negative inputs to the non-selected action values in the action selection system: an exponentially decaying inhibitory current is added at timestep t' as follows:

$$I_{t'}^k = \begin{cases} p_{t'}^k, & \text{if } z_t^k = 1. \\ -\exp\left(\frac{t'-t}{\tau_\phi}\right) p_{t'}^k, & \text{otherwise.} \end{cases} \tag{5}$$

where τ_ϕ controls the leaky speed of the inhibition current for unselected classes. We empirically set $\tau_\phi = 20$ to match the average speech length.

Network resetting. To counter the state saturation problem associated with continual running in RNNs, network state resets are one solution [2, 5], where the challenge is to determine when to reset the network state. Here, we reset the network as a function of when a decision is made *and* when the following empirically derived criterion is satisfied:

- $\min(\bar{a}_t^i) < -\theta$
- $tvar_t^s$ is decreasing and $tvar_t^s - tvar_{t-1}^s < 0.1$.

We see the effect of this condition in the 'Dynamic Resetting' bottom row in Fig. **??**, where the initially incorrect framewise classification in the third sample does not result in a reset, and the sample is correctly classified according to the acc_p measure. Resetting is only applied in the continual running inference phase and not during training.

Metrics. For network accuracy, we measure two metrics: the framewise accuracy acc_f and the prediction accuracy acc_p.

The framewise accuracy for a single sample i is computed as the average accuracy during the network decision process as well:

$$acc_f^i = \frac{1}{t_r^i - t_s^i} \sum_{t=t_s^i}^{t_r^i} \delta(\hat{y}_t^i, y^i), \tag{6}$$

where $\hat{y}_t^i$ is the prediction at timestep $t \in \hat{T}^i$, y^i the correct label for the sample i, and $\delta(\hat{y}_t^i, y^i)$ the Kronecker delta. The average framewise accuracy acc_f is computed as the average over all samples, $\frac{1}{N}\sum_i^N acc_f^i$.

For a sample i, the prediction accuracy acc_p^i is calculated as:

$$acc_p^i = \delta(\hat{y}_{t_r}^i, y^i) \quad \text{iff} \quad |\hat{T}^i| > 10, \tag{7}$$

where $\hat{T}^i$ represents the evidence collection duration calculated as the difference between starting time t_s^i and reset time step t_r^i in sample i, as in Fig. 1. The average prediction accuracy over N samples acc_p is calculated as $acc_p = \frac{1}{N}\sum_i^N acc_p^i$.

Summary. In Algorithm 1, we illustrate the details of the decision-making procedure, including network initialization, network prediction computation, action value calculation based on inhibitory input, dynamic resetting, and metric evaluation.

5 EXPERIMENTS

Dataset. We trained our SRNN on both single training samples from the Google speech dataset v1 (GSCv1) or v2 (GSCv2) [11]. Additionally, we trained a conventional GRU network with an equal number of parameters augmented with the same Temporal Intensity -gating and decision-making structures to provide baseline performance – the GRU network was made up of two densely connected GRU layers with 256 units each. We evaluated these networks by training them to classify all 10 keywords in the GSCv1 and 35 keywords in the GSV2 dataset. Each dataset also contains an additional class for "unknown", and GSV1 also contains a "silence" class. In GSCv1, there are 22,236 training samples and 3,081 test samples, GSCv2 dataset comprises 36,923 training samples and 11,005 test samples. The raw audio is pre-processed via MFCC bandpass filters: each audio sample is passed through 40 2nd order bandpass filters distributed along the Mel-scale between 20Hz and 4kHz. We rescale the response of 40 bandpass filters at each timestep by dividing by the standard deviation across the spectrum. Each keyword sample is converted to a sequence of 101 timesteps in a 40-by-3 matrix representing the spectrum at each time step.

Direct, single sample performance is measured on the respective test datasets. We evaluate the continual running of both the SRNN and GRU on long sequences comprised of concatenations of keywords. To evaluate the network performance on single keyword prediction, we define prediction accuracy acc_p by comparing the prediction and the target when the speech pattern disappears and the network is reset (see also Fig. 1c). We also measure the network performance on long sequences by comparing average frame-wise accuracy acc_f as measured over the whole sequence length. To evaluate the networks' robustness to noise, we applied background noise to each speech audio. Different levels of synthetic noise were applied on the first 10 filter bandpass filters. The noise was generated as Gaussian, $r\mathcal{N}(0,1)$ where r is the noise ratio.

5.1 Results

Performance on single samples. We evaluated the networks on single speech audio samples. For this, we evaluated the GRU network including the Temporal Intensity gating and the decision-making circuit. Then, we compared it to SRNN networks with or without Temporal Intensity gating. Results are shown in Fig. 3: we find that the SRNN with Temporal Intensity gating slightly outperforms the other networks in terms of classification accuracy, including the GRU network, for GSCv1 (Fig. 3a) and GSCv2 (Fig. 3b). Compared to the literature, in [16] ASRNNs achieve 92.14% on GSCv1, slightly better than our dynamic SRNN (89.98%), while the GSCv2 accuracy (87.31%) represents new State-of-The-art (SoTa), exceeding the 79.6% reported in [13].

Noting average activity in the network (Fig. 3c,d), we see that using Temporal Intensity gating lowers the required number of spikes by some 50% for both GSCv1 and GSCv2 tasks. We also find that 68% of spikes in the networks are on average generated during the "active" parts where Temporal Intensity exceeds the signal threshold θ.

Effect of Threshold θ. The parameter θ distinguishes between noise/quiet and speech patterns. Smaller θ will result in noise being more likely treated as part of the speech pattern, while larger θ will cause the network to not identify more words in the recognition process. As such θ directly controls the false positive and false negative rates. In Fig. 4, we plot how θ influences keyword detection as measured in terms of average framewise accuracy acc_f. We see that indeed, as θ increases, the number of missed words grows, and accuracy improves.

Continual running: long sequences. The same networks are also evaluated in the continual-running setting, carrying out continuous inference on speech sequences over longer periods of time. In Table 1, we note the SRNN and GRU networks' performance on concatenated sequences of commands, ranging from a single keyword to 128 concatenated keywords from either GSCv1 or GSCv2. For easy comparison, we report average frame-wise accuracy acc_f when $tvar_t^s > \theta$ for raw output of the network, networks with periodic resetting, and networks with dynamic resetting.

We make several observations from Table 1: first, without resetting both, GRU and SRNN networks saturate, and recognition performance suffers dramatically. Including a periodical reset resolves this issue for the SRNN network (and also for the GRU network, not shown). We then see that our dynamic resetting scheme based on the action selection circuit provides essentially equal (GSCv1) or even slightly better (GSCv2) accuracy.

We also find that with the dynamic resetting mechanism, adding longer silences between concatenated speech samples does not

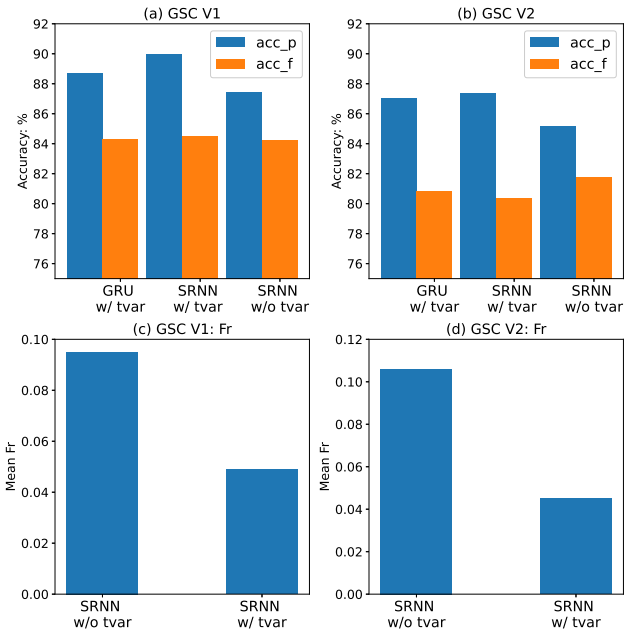

Figure 3: Single sample performance. a) Classification accuracy acc_p and average framewise accuracy acc_f for various baseline networks for GSCv1, and b) for GSCv2. c) average network activity (spike probability per timestep) for GSCv1 and d) GSCv2.

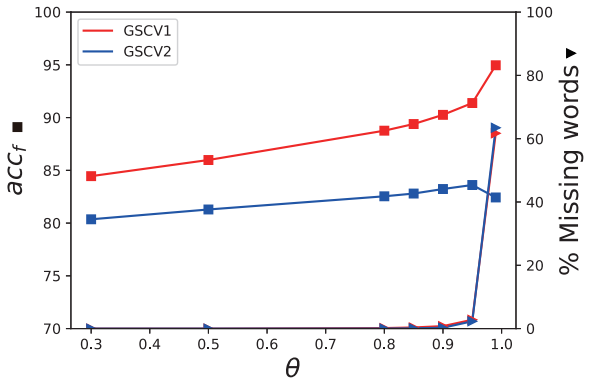

Figure 4: Effect of threshold θ on frame-wise accuracy and % of missing words. The calculation of framewise accuracy only accounts upon detected words.

affect the framewise classification accuracy; an example of such added silence is shown in Fig. 5.

To further quantify the quality of dynamic resetting, we calculate the editing distance [7], both for the GRU and for the SRNN, on concatenated sequences of samples that are correctly classified when presented as single samples. In Fig. 6a, we plot the average number of editing operations needed when evaluating a sequence of 1000 concatenated samples on GSCv2: we find that for both GRU and SRNN, the dynamic reset outperforms the fixed periodic reset

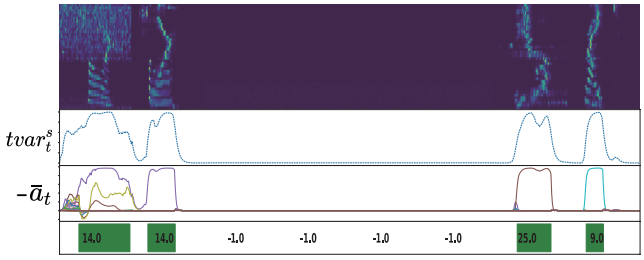

Figure 5: Effect of long salience on SRNN. Label "-1" denotes the added silence audio patches.

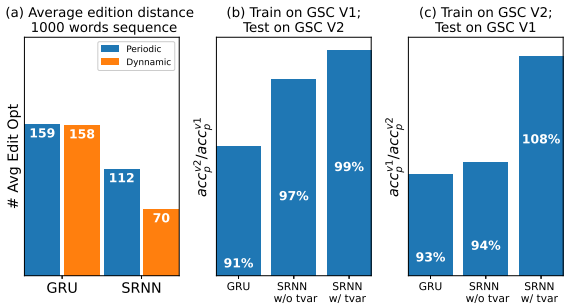

Figure 6: (a) Average editing distance and (b,c) distribution shift robustness computed as percentage of accuracy on the original distribution.

(159 vs 158 for GRU and 112 vs 70 for SRNN); we also find that the SRNN network substantially outperforms the GRU network (70 vs 158 operations).

Temporal Intensity compensates distribution shift. While typical speech benchmarks are comprised of clean samples recorded under essentially ideal conditions, new recordings processed under different circumstances may result in a shifted frequency distribution leading to degraded performance. For example, in [11] a standard CNN model was trained on either of the two versions of GSC datasets and then assessed on both datasets. Depending on the type of CNN, performance was more or less degraded when a network trained on one dataset was evaluated on the other.

Here, we optimized RNNs on either GSCv1 and GSCv2, and evaluated their performance on both datasets. As shown in Fig. 6, we find that standard GRU networks, not gated by Temporal Intensity, show substantial susceptibility to distributions shifts, as average performance drops by 9% (GSCv1 vs. GSCv2) and 7% (GSCv2 vs. GSCv1). For SRNN networks not gated by Temporal Intensity, we find a similar issue; SRNNs with Temporal Intensity -based attention, however, prove to not be sensitive to distribution shift and maintain accuracy (GSCv1 vs. GSCv2) or even improve accuracy (GSCv2 vs. GSCv1, due to the larger training dataset).

6 DISCUSSION

We demonstrated how the inclusion of a local signal-detection measure combined with brain-inspired decision-making circuitry allows compact and high-performance SRNNs to be applied to continual

Dataset	Model	T=1	T=2	T=4	T=8	T=16	T=32	T=64	T=128
GSC V1	GRU raw output	84.31	60.96	47.72	41.37	40.67	37.86	27.98	35.10
	GRU Dynamic Reset	84.31	84.45	83.28	83.74	84.02	82.84	83.07	82.86
	SRNN raw output	83.85	59.54	40.10	27.10	21.28	16.82	15.42	15.03
	SRNN periodicc reset	83.85	83.31	83.07	83.05	82.95	82.98	82.92	82.84
	SRNN action selection with dynamic resetting	84.10	84.09	83.34	83.09	83.03	82.77	82.99	82.88
GSC V2	GRU raw output	80.82	55.25	44.46	39.95	37.59	35.73	35.30	33.02
	GRU Dynamic Reset	80.82	79.58	80.11	79.46	79.77	79.85	79.94	79.82
	SRNN raw output	79.39	48.53	31.45	21.58	15.62	13.49	12.24	11.70
	SRNN periodic reset	79.42	79.38	79.03	79.02	79.31	78.98	78.93	78.88
	SRNN action selection with dynamic resetting	81.73	80.54	80.36	80.12	79.98	80.39	80.05	80.35

Table 1: Average frame-wise accuracy acc_f when $\min\ \tilde{a}_t^i < -\theta$ for different concatenated sequence lengths.

running scenarios. For signals comprised of concatenated keywords, this results in constant-accuracy continual running. Importantly, Temporal Intensity -gating resulted in much reduced average activity in the SRNNs, potentially improving energy consumption. Measured in terms of editing distance, we find that dynamic resetting results in substantially better accuracy, where the SRNN networks outperform GRU networks. We also showed how the decision-making criteria enable the trading-off of false alarms versus missed keywords. A next step will be to evaluate SRNNs on real-world continual running scenarios, which we omitted for lack of a current suitable public benchmark to use and compare to.

We observed furthermore that the Temporal Intensity -gated SRNNs are insensitive to a distribution shift, as measured in terms of environmental transfer performance from GSCv1 to GSCV2 and vice versa. We find this observation somewhat curious, but as noted, similar observations have been made for CNN architectures where some architectures are more or less susceptible to distribution shift.

The absolute classification performance achieved by the SRNN networks is compelling and approaches or exceeds state-of-the-art for SNNs. Still, we believe that the accuracy of the SRNNs can likely be further improved by, for instance, replacing the MFCC features with custom learned ones [8], and optimizing circuit parameters like reset intensity, decision thresholds, and action-selection triggered via lateral inhibition for class specificity. Furthermore, more complex SRNNs can additionally improve sample recognition rates [16], potentially at the expense of increased computational complexity.

Our results open new possibilities in the design of always-on keyword-spotting devices such as the one presented in [6]. This device exploits switched ring oscillators for generating event-based frequency outputs from audio streams. Today, these outputs are processed in a frame-based way using a GRU network. By replacing the frame-based GRU network with SRNNs that directly process event-based features, it will be possible to compute on-demand on the streamed frequency output features, thus further reducing the overall system's power and latency [16] while increasing accuracy.

REFERENCES

[1] Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. [n.d.]. Long short-term memory and Learning-to-learn in networks of spiking neurons. In *NeurIPS 2018*.

[2] Shuo-Yiin Chang, Bo Li, Gabor Simko, Tara N Sainath, Anshuman Tripathi, Aäron van den Oord, and Oriol Vinyals. [n.d.]. Temporal Modeling Using Dilated Convolution and Gating for Voice-Activity-Detection. In *ICASSP2018*.

[3] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. [n.d.]. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning (ICML '06)*.

[4] Kevin Gurney, Tony J Prescott, and Peter Redgrave. 2001. A computational model of action selection in the basal ganglia. I. A new functional anatomy. *Biological cybernetics* 84, 6 (2001), 401–410.

[5] Juntae Kim, Jeehye Lee, and Yoonhan Lee. 2021. Generalizing RNN-Transducer to Out-Domain Audio via Sparse Self-Attention Layers. (Aug. 2021). arXiv:2108.10752

[6] Kwantae Kim, Chang Gao, Rui Graça, Ilya Kiselev, Hoi-Jun Yoo, Tobi Delbruck, and Shih-Chii Liu. 2022. A 23μW Solar-Powered Keyword-Spotting ASIC with Ring-Oscillator-Based Time-Domain Feature Extraction. In *ISSCC2022*, Vol. 65. IEEE, 1–3.

[7] Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. Soviet Union, 707–710.

[8] Tara Sainath, Ron J Weiss, Kevin Wilson, Andrew W Senior, and Oriol Vinyals. 2015. Learning the speech front-end with raw waveform CLDNNs. (2015).

[9] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu. 2016. WaveNet: A generative model for raw audio. *SSW* 125 (2016), 2.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[11] Pete Warden. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209* (2018).

[12] Philipp Weidel and Sadique Sheik. 2021. WaveSense: Efficient Temporal Convolutions with Spiking Neural Networks for Keyword Spotting. (Nov. 2021). arXiv:2111.01456 [cs.LG]

[13] Philipp Weidel and Sadique Sheik. 2021. WaveSense: Efficient Temporal Convolutions with Spiking Neural Networks for Keyword Spotting. *arXiv preprint arXiv:2111.01456* (2021).

[14] Chunyang Wu, Yongqiang Wang, Yangyang Shi, Ching-Feng Yeh, and Frank Zhang. 2020. Streaming Transformer-based Acoustic Models Using Self-attention with Augmented Memory. (May 2020). arXiv:2005.08042

[15] Emre Yılmaz, Özgür Bora Gevrek, Jibin Wu, Yuxiang Chen, Xuanbo Meng, and Haizhou Li. 2020. Deep convolutional spiking neural networks for keyword spotting. In *Interspeech 2020*. ISCA, ISCA.

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

[17] Davide Zambrano, Pieter R Roelfsema, and Sander Bohte. 2021. Learning continuous-time working memory tasks with on-policy neural reinforcement learning. *Neurocomputing* 461 (2021), 635–656.

[18] Yundong Zhang, Naveen Suda, Liangzhen Lai, and Vikas Chandra. 2017. Hello Edge: Keyword Spotting on Microcontrollers. (Nov. 2017).

Algorithm 1: Dynamic resetting

Variables:
- is_rest – network has been reset(1) or not(0)
- h_t – networks' hidden states
- acc_f – framewise accuracy
- acc_p – prediction accuracy
- t_s – start point
- t_r – reset time

Initialization:
- $h_0 = reset(SNN)$
- is_rest=1, $acc_f = acc_p = 0$

for $t = 0$ **to** T **do**
- // **tvar information:**
  - $\mu_t = \|x_t + x_{t-1}\|/2$, $\sigma_t = \|x_t - x_{t-1}\|$
  - $tvar_t = \tanh(\eta\sigma_t\mu_t)$
  - $tvar_t^s = tvar_{t-1}^s + (1-\phi)(tvar_t - tvar_{t-1}^s)$
  - $dtvar_t = tvar_t^s - tvar_{t+1}^s$ – derivative of smoothed tvar
- //**Network prediction:**
  - $p_t = SNN(x_t, tvar_t^s, h_t)$
- //Action selection:
- **for** $i = 1$ **to** n **do**
  - // **Inhibition current :** if detected class is k at t_0
  - **if** *is_reset ==0 and* $i = k$ **then**
    - $I_t^i = p_t^i$
  - **end**
  - **if** *is_reset ==0 and* $i \neq k$ **then**
    - $I_t^i = -\exp\left(\frac{t-t_0}{\tau_\phi}\right) p_t^i$
  - **end**
  - // **Action value:**
  - $u_t^i = -w_z^- I_t^i + w_z^+ \sum_{j\neq i}^n I_t^j$
  - $a_{t+1}^i = a_t^i + (1-\rho)(u_t^i - a_t^i)$
  - $\bar{a}_t^i = \tanh(tvar_t^s a_t^i)$
- **end**
- Prediction: $z_t = \underset{i\in 1,2,\ldots n}{\arg\min}(\bar{a}_t^i)$
- // **Decision making:**
- **if** $\min(\bar{a}_t) < -\theta$ *and* $dtvar_t > 0$ *and is_rest ==1* **then**
  - // Start decision collection, detected class k at t_0
  - // rest statue
  - is_reset = 0 ; t_s = t
- **end**
- // **Framewise Accuracy:** compare current prediction z_t and target label y
- **if** *is_reset ==0* **then**
  - $acc_f += (z_t == y_t)$
- **end**
- **if** $\min(\bar{a}_t) < -\theta$ *and* $dtvar_t < 0$ *and is_rest ==0* **then**
  - // End decision collection and reset hidden states
  - $h_t = reset(SNN)$
  - **if** $(\hat{T} = t - t_s) > 10$ **then**
    - // **Prediction Accuracy:** compare final prediction z_t and target label y
    - $acc_p += (z_t == y_t)$
  - **end**
  - // rest statue
  - is_reset = 1
- **end**

end