Neuromorphic visual attention for Sign-language recognition on SpiNNaker

Šárka Lísková[1*], Olha Vedmedenko[2*], Mazdak Fatahi[2], Matej Hoffmann[1], P. Michael Furlong[3], and Giulia D'Angelo[1]

[1]Dept. of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic
[2]Faculty of Information Technology, Czech Technical University in Prague, Czech Republic
[3]Université de Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
[4]National Research Council of Canada & Systems Design Engineering, University of Waterloo, Canada

***Abstract*—Sign-language recognition has achieved substantial gains in classification accuracy in recent years; however, the latency and power requirements of most existing methods limit their suitability for real-time deployment. Neuromorphic sensing and processing offer an alternative paradigm based on sparse, event-driven computation that supports low-latency and energy-efficient perception.**

In this work, we introduce an end-to-end neuromorphic architecture for American Sign Language (ASL) fingerspelling recognition that integrates a spiking visual attention mechanism for online region-of-interest extraction with a compact spiking neural network deployed on the SpiNNaker neuromorphic platform. We benchmark the proposed system against two datasets: a synthetically generated event-based version of the Sign Language MNIST dataset and a natively recorded ASL-DVS dataset, whilst providing a comprehensive overview of Sign-language recognition and related work.

This work yields competitive performance in simulation (92.27%) and comparable performance on neuromorphic hardware deployment (83.1%), while achieving the most energy-efficient architecture (0.565 mW) and low latency (3 ms) across all benchmarked approaches. Despite its compact design, the system demonstrates the suitability of task-dependent visual attention applications for edge deployment.

***Index Terms*—Visual attention, active object recognition, Sign-language recognition, event-based sensing, and neuromorphic computing**

I. Introduction

Natural hand gestures, including Sign-language, play an important role in human communication, particularly for deaf and hard of hearing users. Recent literature focuses primarily on accuracy of Sign-language recognition [1]; however, for interaction to be effective and reactive, gesture-recognition systems must achieve low latency, low power consumption, and high robustness to environmental variability. Latencies exceeding 100-200 ms break the illusion of agency [2], [3], leading to user frustration and reduced trust. Consequently, Sign-language recognition in interactive settings requires perception and decision-making to be continuously responsive, requiring lightweight and power-efficient systems suitable for deployment on robotic platforms or wearable devices.

*These authors contributed equally. Corresponding authors: sarka.liskova@fel.cvut.cz, vedmeolh@fit.cvut.cz

In humans, visual attention mechanisms enable selective processing of task-relevant regions of interest (ROIs), focusing perceptual resources toward salient scene elements while filtering irrelevant information [4]. This offers a bio-inspired strategy for achieving both efficiency and responsiveness. By contrast, modern, data-driven systems [5] that process the entire visual field uniformly allocate computational resources to non-informative regions of the scene, compromising both efficiency and robustness.

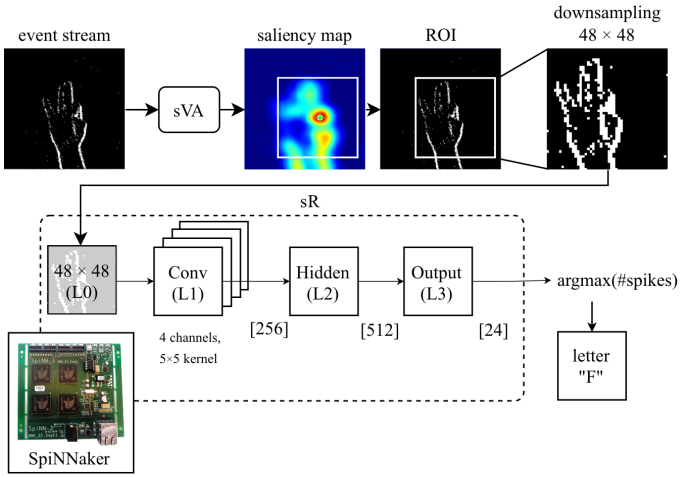

Fig. 1. **Neuromorphic Sign-language recognition architecture:** Event streams are processed through the spiking visual attention (sVA) model to extract salient regions (ROIs), which are then downsampled and fed into the recognition network (sR) deployed on SpiNNaker. The network architecture comprises convolutional (L1, 256 neurons, 4 channels, 5×5 kernel), hidden (L2, 512 neurons), and output (L3, 24 output neurons) layers.

Without attentional selectivity, these approaches incur significant computational burdens [5] and exhibit weaknesses in generalisation across varying scales, orientations and illumination conditions [6]. Biologically inspired frame-based attention mechanisms [7]–[9] can alleviate these limitations by reducing the volume of data requiring detailed analysis, focusing computational resources on ROIs that contain the relevant information in the scene.

However, these frame-based bio-inspired visual attention models [7]–[9] rely on dense RGB frames, which are prone to

motion blur, limited dynamic range, and high data throughput. By contrast, event-based or Dynamic Vision Sensors (DVS) [10], [11] achieve efficiency at the hardware level through asynchronous, sparse encoding of visual changes.

Inspired by the mammalian retina, DVS devices asynchronously report per-pixel illumination changes with microsecond temporal resolution and signed polarity. This event-driven mechanism eliminates motion blur, increases invariance to illumination changes, extends dynamic range, and reduces latency. By transmitting only spatiotemporal changes rather than complete frames, the data throughput, power consumption, and storage requirements are reduced. These advantages make event-based cameras particularly suited for real-time applications.

Event-driven bio-inspired saliency-based visual attention models [12]–[14] address the data acquisition challenges of conventional vision systems, achieving processing latencies of $\sim$100 ms for event-based attention [12]. However, realising their full potential requires a corresponding shift in computational processing, as frame-based processing of event data introduces a fundamental mismatch between asynchronous sensing and synchronous computation. Neuromorphic hardware [15], whether analog [16]–[18] or digital [19]–[21], eliminates the von Neumann bottleneck by co-locating memory and processing, enabling Spiking Neural Networks (SNNs) to process asynchronous event streams in a massively parallel fashion [22]–[25], with activity-scaled power consumption mirroring biological neural efficiency, making them suited for real-time applications [26], [27].

These gains are demonstrated concretely by event-based saliency models, which achieve a 6$\times$ latency reduction (from $\sim$100 ms [12] to $\sim$16 ms [28]) when migrated to SpiNNaker neuromorphic hardware. Building on SNN saliency-based foundations [28], D'Angelo et al. [29] developed an end-to-end spiking-based, learning-free visual closed-loop exploration architecture on a pan-tilt unit, achieving competitive salient object detection across diverse lighting conditions with $\sim$124 ms latency using Metal Performance Shaders.

However, all the aforementioned works do not integrate attention mechanisms with concrete recognition tasks, limiting their applicability to goal-directed visual processing.

Gesture recognition represents one such task where this integration would be beneficial. Existing approaches process DVS data using either conventional artificial neural networks (ANNs) [30] or SNNs [31]–[33], often not deployed on dedicated neuromorphic hardware and instead relying on conventional CPUs or GPUs. More recent work has demonstrated gesture recognition systems deployed directly on neuromorphic platforms [34]–[40], exploiting the full potential of neuromorphic computing (See the final Table VII for a detailed comparison). Therefore, the majority of DVS datasets focus on dynamic gestures, as the motion information captured by event cameras provides advantages over conventional RGB cameras for tracking rapid movements.

In contrast, few datasets address static signs and hand shapes, such as Sign-language fingerspelling alphabets, which constitute an important component of Sign-language communication. State-of-the-art fingerspelling recognition is dominated by frame-based RGB cameras and with deep ANNs, operating on images or video clips [41]. Although RGB-based deep learning achieves 90–98% accuracy on benchmarks of static fingerspelling or continuous signing sequences [42], it remains ill-suited to deployments where responsiveness and energy efficiency are as critical as recognition accuracy. To the best of the authors' knowledge, ASL-DVS [43] is the only natively recorded DVS dataset focused on static fingerspelling, with recognition performed on a GPU; although there are natively recorded DVS datasets of dynamic Sign-language gestures available [31]. The rest of the fingerspelling datasets consist of frame-based RGB datasets, such as the Sign Language MNIST (SL-MNIST) [44], or the ASL fingerspelling Dataset [45]. Therefore, no existing end-to-end spiking-based fingerspelling recognition system leverages neuromorphic computing to fully exploit its inherent low-latency and low-power advantages.

To address this gap, we propose an architecture for fingerspelling recognition in American Sign Language (ASL) that leverages end-to-end spiking attention-driven ROI selection to reduce input data, followed by fingerspelling classification on the neuromorphic platform SpiNNaker [19], which minimizes computational loads through energy-efficient neuromorphic processing. This work targets static fingerspelling, leveraging neuromorphic capabilities to capture individual letter poses before the transient event data decays, demonstrating ultra-low-latency efficiency for real-time applications.

The contributions of this work are listed below:

1) System-level integration of a lightweight, spiking-based, learning-free visual attention model [29] that selectively allocates computational resources to task-relevant regions and effectively filters background clutter with a spiking-based fingerspelling recognition model.

2) Design of an efficient and compact neuromorphic architecture for recognition deployed on the SpiNNaker neuromorphic platform [19] (SpiNN-3), explicitly constrained by the hardware resource limits (neurons, synapses, routing).

3) Benchmark evaluation on two fingerspelling datasets: Sign Language MNIST [44] and ASL-DVS [46], where the former is converted to events using a standard pipeline and made available for reproducible benchmarking (Section V).

4) Validation through both simulation and hardware deployment on SpiNNaker.

5) System characterization of latency and power consumption measured on the SpiNNaker platform.

6) Comprehensive comparison of state-of-the-art gesture and fingerspelling recognition systems (final comparison Table VII): accuracy, inference latency, and power consumption.

II. Methods

A. spiking-based Visual Attention model

This work (see Figure 1) extends the spiking-based visual attention model (sVA) proposed by D'Angelo et al. [28],

[29] to process asynchronous event streams from event cameras. The sVA model employs a bio-inspired saliency mechanism [7] to extract relevant bottom-up information from the visual field. This model takes inspiration from border ownership cells in the visual cortex [47], which perceptually group information to identify salient regions that potentially contain objects. The model emulates this functionality by pooling the information from several orientations of the curved von Mises filter (see Equation 1), where x and y are the kernel coordinates with the origin at the center of the filter, θ represents its orientation, I_0 is the modified Bessel function of the first kind, R_0 is the radius of the filter representing the distance of the VM filter from the center, and $\tan^{-1}$ takes two arguments returning values in radians in the range $(-\pi, \pi)$.

$$VM_\theta(x,y) = \frac{\exp(\rho \cdot R_0 \cdot \cos(\tan^{-1}(-y,x) - \theta)}{I_0(\sqrt{x^2+y^2-R_0})} \quad (1)$$

This radius will later define the radius of the opposite orientation of the von Mises filters, grouping information to detect the proto-object and defining the range of detectable sizes. Our approach employs a two-stage pipeline: the sVA extracts regions of interest from event streams, which are then processed by a compact feed-forward spiking neural network (sR) for Sign-language fingerspelling recognition.

B. spiking Recognition model

The sR model processes the extracted visual attention ROIs, operating on downsampled 48×48 event inputs, a resolution empirically determined to be sufficiently large to preserve relevant visual information while satisfying SpiNN-3 resource constraints. The network architecture (see Table I) consists of a single convolutional layer (L1) (4 channels, 5×5 kernel) of 256 neurons for spatial feature extraction, followed by a fully connected Hidden layer (L2) of 512 neurons and a linear readout Output layer (L3) with 24 output neurons representing the 24 static letters of the ASL fingerspelling alphabet, with L2 and L3 using LIF dynamics and classification determined by argmax over output spike counts.

The architecture is shaped by SpiNN-3 hardware constraints: The size of L3 is fixed by the number of classes, while L1 is required to spatially downsample the input to a tractable representation within SpiNNaker's connectivity limits by mapping the 48×48 input into 4 channels, each represented by an 8×8 feature map, yielding a total of 256 neurons, while the 5×5 kernel size ensures spatial selectivity without incurring excessive synaptic connectivity. The size of L2 (512 neurons) is selected to maximally utilise the remaining neuron and synapse budget. The number of timesteps is likewise constrained by the real-time latency objective. The presented architecture reflects a hardware-feasible trade-off operating point consistent with the goal of demonstrating a real-time, on-edge neuromorphic system.

III. Experiments & Results

The system architecture is validated through simulation using snnTorch [48] and hardware deployment on SpiNNaker.

TABLE I
Network architecture and connectivity parameters.

Population	# Neurons	Input Syn.	Connection
Input (L0)	2304 (48×48)	—	—
Conv (L1)	256	6400	Conv.
Hidden (L2)	512	130938	Fully conn.
Output (L3)	24	12284	Fully conn.
Total	3096	149622	

Event streams are processed by the sVA on a MacBook with an Apple M4 Pro processor, reserving SpiNNaker's computational resources entirely for the sR. Although the sVA runs on a conventional processor in this work, the pipeline is fully spiking and neuromorphic-compatible: the lightweight sVA model has been previously validated on SpiNNaker neuromorphic hardware for real-time operation [28], and the current split reflects the neuron and synapse budget constraints of the SpiNN-3 board rather than an architectural limitation. The sVA model achieves 88.8% accuracy for salient object detection (against ground truth segmentation masks) in office scenarios and 89.8% in challenging indoor and outdoor low-light conditions [29] on the event-assisted low-light video object segmentation dataset [49]. The ROI extracted by sVA is then fed into the sR model for the sign-letter classification. We tested the sR model in two conditions: in simulation on an Apple processor using the Metal Performance Shaders, and the other running the spiking model directly on the SpiNNaker.

A. Event-based Sign-language datasets

The event-based vision community has predominantly focused on gesture recognition [30], [35], [38], providing many DVS dynamic gesture datasets, such as the IBM DVSGesture Dataset [36] or the SL-Animals-DVS Dataset [34], but the availability of native DVS datasets for Sign-Language fingerspelling (static signs) remains limited. To validate the sR model, we evaluate our approach on two ASL fingerspelling datasets: Sign language MNIST dataset (SL-MNIST) [44] and the ASL-DVS [46]. Sign Language MNIST allows comparison with existing SNN architectures deployed on Loihi [50] using the same benchmark, while ASL-DVS is, to the best of our knowledge, the only natively event-based fingerspelling dataset available.

1) Sign Language MNIST: The Sign Language MNIST dataset (SL-MNIST) [44] is a widely-used benchmark for ASL fingerspelling recognition, consisting of 34627 grayscale 28×28 images of 24 static letters (J and Z excluded because of dynamic nature of the gestures), divided into 27455 training and 7172 test samples. Preprocessing included cropping, grayscale conversion, resizing, and data augmentation via filtering, pixelation, and rotation. The hand isolation step introduces computational overhead incompatible with real-time low-latency operation.

To test our architecture, events are generated by applying a random walk pixel shift to the static grayscale images,

creating synthetic video sequences that are subsequently converted to event streams using the IEBCS (Image-to-Event-Based-Camera Simulator) framework [51]. The conversion was performed once with a fixed random seed, producing a static event-based dataset with no run-to-run variance. These event streams are processed by the lightweight sVA model to extract ROIs in a computationally efficient manner suitable for online demonstration, as previously shown by D'Angelo et al. [29]. For experimental consistency, we adopted the original train–test split defined by the authors of the SL-MNIST dataset.

2) ASL-DVS dataset: The ASL-DVS dataset (ASL-DVS), introduced by Bi et al. [46], is a large natively recorded event-based dataset for Sign-language fingerspelling recognition under real-world conditions. The dataset was acquired using an iniLabs DAVIS240c neuromorphic vision sensor positioned in an office environment with low environmental noise and constant illumination. It contains over 100000 samples across 24 classes, like Sign Language MNIST; it excludes letters J and Z. Five subjects were recorded performing the different static handshapes relative to the camera to introduce natural variance into the dataset. For each letter class, 4200 samples were collected, with each sample capturing approximately 100 ms of asynchronous event data. Following this subject-wise organization, we adopted a cross-subject evaluation protocol, using data from four signers for training and validation and reserving the remaining signer exclusively for testing.

B. Role of spiking-based Visual Attention

Figure 2 illustrates the need for the sVA module on ROI selection, contrasted against a naive center-crop baseline. Since subjects vary their hand position freely, center-cropping frequently captures background rather than the hand, while sVA reliably localises the salient region regardless of position.

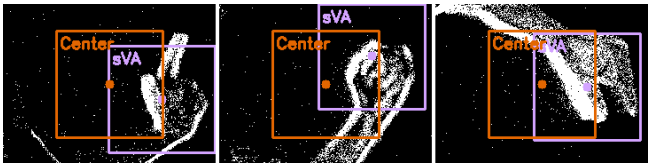

Fig. 2. **sVA ROI (purple) vs. naive center crop (orange)** on three ASL-DVS frames (letters L, O, Q).

C. Simulation

The sR model, described in Table I, is implemented using snnTorch [48] and simulated on a MacBook with an Apple M4 Pro processor. All neurons of L2 & L3 follow LIF dynamics with fixed membrane decay $\beta = 0.92$ (See parameters in Table II). For both datasets, each input sample was simulated for 35 discrete timesteps to align with the temporal extent of the event-based window data, as in [46]. Additionally, given the heavy downsampling of the original SL-MNIST dataset, we explored an extended timestep of 100 ms to investigate whether prolonged temporal integration could improve model performance.

TABLE II
TRAINING CONFIGURATION ON SNNTORCH.

Parameter	SL-MNIST		ASL-DVS
Leak factor (β)	0.92	0.92	0.92
Time steps	100	35	35
Epochs	310	130	170

The network is trained end-to-end via surrogate gradient backpropagation (See Table III) through time with a fast-sigmoid surrogate function, using AdamW optimization (learning rate 10^{-3}, weight decay 10^{-4}), Kaiming initialisation for convolutional and Xavier initialization for fully connected layers. To address class imbalance and improve performance on hard-to-classify samples (such as letter "U" against "V" and "M" against "N", etc.), training employed focal loss with label smoothing (ϵ), class-balanced weighting, and online hard example mining. The learning rate followed a cosine annealing schedule with warm restarts to improve convergence stability (See Table III).

TABLE III
TRAINING OPTIMIZATION PARAMETERS ON SNNTORCH.

Component	Parameters
Loss	Focal: $\gamma = 2.0$, $\varepsilon = 0.1$, mining thr: 0.65
Optimizer	AdamW: lr 3×10^{-3}, $\beta = (0.9, 0.999)$, decay 10^{-4}
Scheduler	Cosine: $T_0 = 25$, $T_{\text{mult}} = 2$, $\eta_{\min} = 2 \times 10^{-7}$

Classification is performed by temporally accumulating output spikes and selecting the class with maximal spike count. The bias-free, compact architecture enables direct deployment on neuromorphic hardware platforms such as SpiNNaker.

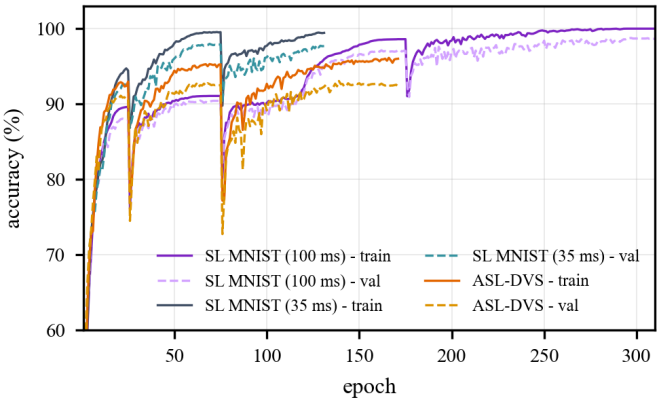

Fig. 3. **Training and validation accuracy curves** for SL-MNIST dataset (35 ms and 100 ms samples) and for the ASL-DVS dataset (35 ms samples).

Figure 3 shows the learning dynamics across both datasets. In both cases, the sR model exhibits rapid initial learning, demonstrating efficient adaptation to event-based Sign-language fingerspelling data. The SL-MNIST dataset shows

faster initial convergence compared to ASL-DVS, likely due to its synthetic generation via random-walk pixel shifts, whereas ASL-DVS captures natural human motion. Training and validation curves indicate consistent learning without significant overfitting. Table V reports test set performance evaluated in simulation. SL-MNIST achieves 74.61% accuracy for 35 ms and 83.82% accuracy for 100 ms, while ASL-DVS reaches 92.27%, both showing performance gaps compared to validation results in Figure 3. The validation–test gap on SL-MNIST is attributed to stochastic variability introduced by the random-walk–based RGB-to-event conversion, which affects event timing and sparsity despite identical preprocessing across train, validation and test splits. Increasing the temporal integration window from 35 ms to 100 ms partially alleviates this effect by enabling additional temporal evidence accumulation, indicating that event sparsity and temporal variability are key contributors to the reduced test accuracy on SL-MNIST. By contrast, the natively recorded ASL-DVS dataset captures real human hand motion with real DVS sensors, resulting in richer event patterns and a smaller generalization gap and higher test accuracy. Additional differences between the datasets, including the lower spatial resolution of the original SL-MNIST images and the upsampling applied prior to event conversion, may further contribute to the observed performance disparity relative to natively recorded DVS data. In the original work by Bi et al. [43], the event-based ASL-DVS dataset is addressed using graph-based event representations and graph and residual-graph CNN architectures. In contrast, we adopt a compact spiking network consisting of a single convolutional layer followed by fully connected stages. Despite its reduced architectural complexity, our model achieves an accuracy of 92.2%, which is improved compared to the reported performance of the G-CNN (87.5%) and RG-CNN (90.1%). These results indicate that competitive recognition performance can be obtained using a compact spiking architecture.

D. SpiNNaker deployment

The sR model was then deployed on a SpiNN-3 [19] development board containing 4 SpiNNaker chips, each equipped with 18 ARM968E-S cores. The membrane time constant τ_m is derived from the leak factor β using the relationship $\tau_m = -\Delta t/\ln(\beta)$. Excitatory and inhibitory synaptic time constants (τ^E_{syn}, τ^I_{syn}) control synaptic memory dynamics, while the refractory period τ_{refrac} provides burst suppression. Weight scaling factors (w_{fc}, w_{out}) and inhibitory weights (w_{inh}) modulate connection strengths between layers (See Table IV for the parameters of the deployed network).

1) Accuracy: Table V shows the accuracy values on SpiNNaker for the SL-MNIST and the ASL-DVS dataset as 71.7% and 83.1%, respectively. Only the 100 ms SL-MNIST model was deployed on hardware, for two reasons: it enables a fair comparison with ASL-DVS, and the 35 ms simulation with an accuracy of 74.61% is too low to yield a meaningful hardware result given the known simulation-to-hardware gap. The observed performance gap between simulation and deployment is primarily attributed to weight quantization: SpiNNaker uses fixed-point weight representation versus the 32-bit floating-point precision of snnTorch, reducing the effective resolution of synaptic strengths. Additional factors include differences in synaptic dynamics and the more biologically realistic LIF formulation employed on SpiNNaker. Quantization-aware training and precision-aware calibration [52] are planned for future work to reduce this discrepancy. Despite these differences, the overall behavior of the deployed network closely follows the simulation results, indicating that the trained model transfers effectively to the neuromorphic hardware platform.

TABLE IV
SPINNAKER DEPLOYMENT PARAMETERS.

Parameter	Value	Parameter	Value
Δt (ms)	1.0	τ^E_{syn} (ms)	5.0
d (ms)	1.0	τ^I_{syn} (ms)	3.0
τ_m (ms)	$-\Delta t/\ln(\beta)$	τ_{refrac} (ms)	1.0
V_{rest} (mV)	-65	w_{fc}	0.3
V_{reset} (mV)	-65	w_{out}	2.0
V_{thresh} (mV)	-61	w_{inh}	1.0

TABLE V
SR TEST SET PERFORMANCE IN SIMULATION AND ON SPINNAKER.

Platform	Accuracy SL-MNIST 35 ms	Accuracy SL-MNIST 100 ms	Accuracy ASL-DVS 35 ms
Mac (simulation)	74.61%	83.82%	92.27%
SpiNNaker	–	71.7%	83.1%

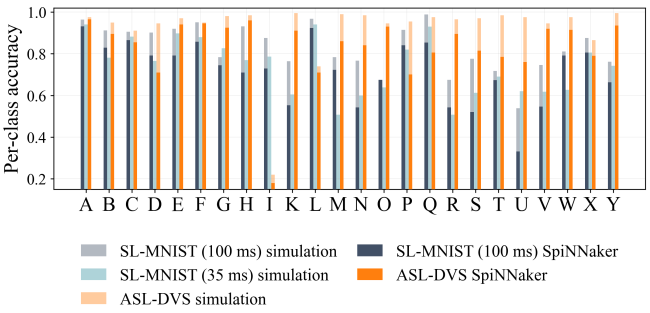

Fig. 4. **Comparison of per-class test accuracy of the sR model evaluated in simulation and deployed on SpiNNaker** for the SL-MNIST and the ASL-DVS dataset. The horizontal axis indicates the ASL fingerspelling classes, while the vertical axis reports classification accuracy (%).

Figure 4 shows the per-class simulation and after-deployment accuracy for each letter of the ASL alphabet, demonstrating consistent performance across the native event-based ASL-DVS dataset, with the exception of letter "I", which exhibits notably lower accuracy likely due to visual similarity with other signs. Both simulated and deployed models show consistent performance between 35 and 100 ms temporal windows of SL-MNIST dataset samples, with the longer duration yielding superior results, indicating that the model benefits from extended temporal integration.

2) Latency and Power consumption: To demonstrate the advantages of neuromorphic hardware, we report the latency

and power consumption of the network deployed on the SpiNNaker platform (See Table VII for final comparison). The network architecture comprises two sequential layers: input-to-hidden and hidden-to-output connections. Given SpiNNaker's fixed internal digital clock of $1\ ms$, the total inference latency is $3\ ms$ (one time step per layer plus input processing).

Following standard SpiNNaker power models [53], [54], total power is given by $P_T = P_I + P_B + (P_N \times n) + (P_S \times s)$, where P_I is idle power, P_B is baseline power for system services, P_N is per-neuron power, and $P_S \approx 8$ nJ is the energy per synaptic event. We focus on dynamic power ($P_S \times s$) to quantify neuromorphic processing efficiency independent of system overhead. Table VI presents the average number of synaptic events per layer, calculated by dividing the total spike count by 120 samples, where each sample comprises the events collected over a 35 timesteps (where each $timestep = 1\ ms$) for each Sign-language gesture. The estimated energy consumption is approximately 24891.84 nJ and 19807.68 nJ, corresponding to a power consumption of approximately 0.711 mW for the SL-MNIST and 0.565 mW for the ASL-DVS dataset, as shown in Table VI. sVA latency, validated on MPS in [29], is 68.29 ms $\pm$ 6.19 ms on ASL-DVS dataset (M4 MPS).

TABLE VI
POWER CONSUMPTION MEASUREMENTS

Layer	SL-MNIST Avg Spikes	ASL-DVS Avg Spikes
Conv (L1)	719.31	720.95
Hidden (L2)	2235.82	1681.92
Output (L3)	156.35	73.09
Total	3111.48	2475.96
Energy consumption	$\approx$ 24891.84 nJ	$\approx$ 19807.68 nJ
Power consumption	$\approx$ 0.711 mW	$\approx$ 0.565 mW

IV. BENCHMARK

Table VII summarizes recent work on gesture recognition and Sign-language fingerspelling recognition, with emphasis on modality (DVS/RGB), hardware platform, and reported performance metrics. To provide an overview of research in this specific application domain, we also include gesture recognition studies. While a substantial body of work has addressed dynamic gesture recognition using DVS data, and some models have been deployed on neuromorphic hardware such as TrueNorth [34], [36], Loihi [35], [37], [40], or SpiNNaker [38], [39], only a small subset of studies focus on and report inference latency and power consumption [36]–[39]. The overall accuracy across dynamic gesture recognition models is reported to be typically high, exceeding 90% in most cases. However, the majority of reported (or absent) latency and power consumption values in Table VII suggest that classification accuracy is predominantly prioritized, with comparatively less emphasis on end-to-end efficiency metrics.

In contrast, neuromorphic work focusing specifically on Sign-language fingerspelling recognition is scarce. Existing SNN-based approaches for sign and gesture recognition typically employ multiple convolutional layers followed by large fully connected stages, often comprising several hundred hidden units [31], [34], [35]. While such architectures can achieve high classification accuracy [33], [36], their depth and parameter count pose challenges for extremely low-latency and energy efficiency.

A comparable architecture in terms of model compactness is the multilayer perceptron (MLP) proposed by Mohammadi et al. [50], consisting of 1576 neurons, which was trained for SL-MNIST recognition and deployed on the Intel Loihi platform. While this model achieves higher classification accuracy (90.69%), it operates at a substantially higher inference latency (12.64 ms) and dynamic power consumption (61.37 mW). In contrast, our proposed sR model achieves lower accuracy on the same SL-MNIST benchmark, but reduces inference latency to 3 ms, 4x times reduced and dynamic power consumption 86x times. This comparison highlights the different operating points of the two approaches and illustrates the emphasis of the proposed system on ultra-low-latency and energy-efficient deployment. The lightweight sR model proposed in this work reports the lowest power consumption among existing Sign-Language recognition systems. On the ASL-DVS real-world DVS data benchmark, it achieves slightly lower but comparable accuracy in hardware deployment, whilst maintaining performance parity in simulation, despite its extremely compact architecture. Moreover, the fully event-driven pipeline supports online, continuous inference with minimal latency, establishing a foundation for future real-time applications beyond the scope of this work.

V. CODE AND DATA AVAILABILITY

The open-source architecture is available at [1]. Data of the event-based SL-MNIST can be found here [2].

VI. CONCLUSION

This work proposes an end-to-end neuromorphic spiking-based pipeline that couples spiking visual attention with low-latency on-chip fingerspelling recognition on SpiNNaker [19] neuromorphic platform, demonstrating that effective ASL fingerspelling recognition is achievable under strict hardware resource constraints. Across both benchmark datasets [43], [44], the system reports the lowest power consumption and among the lowest latencies of all compared approaches, while achieving competitive accuracy in simulation and comparable, though slightly reduced, accuracy in hardware deployment.

The results demonstrate that competitive recognition performance can be achieved with substantially reduced architectural complexity compared to prior SNN-based gesture and sign recognition approaches, while explicitly optimizing the end-to-end system for deployability on neuromorphic hardware.

[1] https://github.com/Neuro-inspired-Perception-and-Cognition/Sign-language-Recognition

[2] https://drive.google.com/drive/folders/1ti0ou-iGFHyBpfTag3ftHcD25nXTtrfj?usp=sharing

Ref	Dynamic Gestures	ASL Fingerspelling	Sensor	Network	Hardware	Acc.	Latency (ms)	Power Usage (mW)
[31]	DVS_Sign	-	DVS	SNN	CPU, GPU	77%	-	-
[32]	11 Kinect gestures	-	stereo DVS	SNN + HMM	CPU	>90%	-	-
[33]	DVSGesture IBM	-	DVS	LSM + CNN	CPU + GPU	**98.42%**	-	-
[30]	DVSGesture IBM	-	DVS	TCN ANN	Kraken PULP SoC, CUTIE accelerator	97.7%	**0.9**	4.7
[34]	Gestures for ASL animals (19 signs)	-	DVS	SNN (SLAYER, STBP)	TrueNorth	-	-	-
[36]	DVSGesture IBM	-	DVS	conv SNN	TrueNorth	96.5%	**105** avg	178.8
[35]	5 gestures + EMG	-	DVS + EMG	SNN	Intel Loihi	-	-	-
[37]	DVSGesture IBM	-	DVS	DNN-trained SNN	Intel Loihi	89.64%	161.42	-
[40]	DVSGesture IBM	-	DVS	SNN	Intel Loihi	-	-	-
[38]	DVSGesture IBM	-	DVS	SNN	SpiNNaker2	94%	-	459 mJ
[39]	DVSGesture IBM	-	DVS	EGRU SNN	SpiNNaker2	**96.83%**	-	390
[55]	-	24 ASL Signs	DVS	ANN	FPGA	79.58%	-	-
[50]	-	SL-MNIST	rate-based encoded RGB	SNN VGGNet SNN MLP	Intel Loihi	**97.7%** 90.69%	8.03 12.64	71.89 61.37
[43]	-	ASL-DVS	DVS	G-CNN RG-CNN	GPU	87.5% 90.1%	-	-
sR (this work)	-	SL-MNIST	RGBtoDVS	SNN	SpiNNaker [19]	71.7%	**3**	**0.711**
sR (this work)	-	ASL-DVS	DVS	SNN	SpiNNaker [19]	83.1%	**3**	**0.565**

TABLE VII

SUMMARY OF RECENT WORK IN DVS GESTURE RECOGNITION. ABBREVIATIONS: TERNARIZED CONVOLUTION NETWORK (TCN), HIDDEN MARKOV MODEL (HMM), ARTIFICIAL NEURAL NETWORK (ANN), SPIKING NEURAL NETWORK (SNN), LIQUID STATE MACHINE (LSM), DEEP NEURAL NETWORK (DNN), EVENT-BASED GATED RECURRENT UNIT (EGRU).

The combination of real-time attention-driven ROI selection, high processing speed, low power consumption, and competitive accuracy suggests the potential of deploying this compact network in wearable devices and edge computing applications.

ACKNOWLEDGMENTS

GDA was supported by the European Union's HORIZON-MSCA-2023-PF-01-01 research and innovation programme under the Marie Skłodowska-Curie grant agreement ENDEAVOR No. 101149664. SL and MH were supported by the project ROBOPROX (reg. no. CZ.02.01.01/00/22_008/0004590). SL and MH were additionally supported by the Czech Science Foundation (GAČR) project PIONEER, no. 26-23432S. We thank Bedrich Himmel for technical assistance.

REFERENCES

[1] J. Li, J. Zhong, and N. Wang, "A multimodal human-robot sign language interaction framework applied in social robots," Frontiers in neuroscience, vol. 17, p. 1168888, 2023.

[2] S. K. Card, G. G. Robertson, and J. D. Mackinlay, "The information visualizer, an information workspace," in Proceedings of the SIGCHI Conference on Human factors in computing systems, pp. 181–186, 1991.

[3] R. B. Miller, "Response time in man-computer conversational transactions," in Proceedings of the December 9-11, 1968, fall joint computer conference, part I, pp. 267–277, 1968.

[4] G. Rizzolatti, "Mechanisms of selective attention in mammals," in Advances in vertebrate neuroethology, pp. 261–297, Springer, 1983.

[5] A. Gizdov, S. Ullman, and D. Harari, "Seeing more with less: Human-like representations in vision models," in Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 4408–4417, 2025.

[6] H. Zhu, H. Wei, B. Li, X. Yuan, and N. Kehtarnavaz, "A review of video object detection: Datasets, metrics and methods," Applied Sciences, vol. 10, no. 21, p. 7834, 2020.

[7] A. F. Russell, S. Mihalaş, R. von der Heydt, E. Niebur, and R. Etienne-Cummings, "A model of proto-object based saliency," Vision research, vol. 94, pp. 1–15, 2014.

[8] J. L. Molin, A. F. Russell, S. Mihalas, E. Niebur, and R. Etienne-Cummings, "Proto-object based visual saliency model with a motion-sensitive channel," in 2013 IEEE Biomedical Circuits and Systems Conference (BioCAS), pp. 25–28, IEEE, 2013.

[9] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," IEEE Transactions on pattern analysis and machine intelligence, vol. 20, no. 11, pp. 1254–1259, 1998.

[10] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128$times$128 120 db 15μ s latency asynchronous temporal contrast vision sensor," IEEE journal of solid-state circuits, vol. 43, no. 2, pp. 566–576, 2008.

[11] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, et al., "Event-based vision: A survey," IEEE transactions on pattern analysis and machine intelligence, vol. 44, no. 1, pp. 154–180, 2020.

[12] M. Iacono, G. D'Angelo, A. Glover, V. Tikhanoff, E. Niebur, and C. Bartolozzi, "Proto-object based saliency for event-driven cameras," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 805–812, IEEE, 2019.

[13] S. Ghosh, G. D'Angelo, A. Glover, M. Iacono, E. Niebur, and C. Bartolozzi, "Event-driven proto-object based saliency in 3d space to attract a robot's attention," Scientific reports, vol. 12, no. 1, p. 7645, 2022.

[14] G. D'Angelo, S. Voto, M. Iacono, A. Glover, E. Niebur, and C. Bartolozzi, "Event-driven figure-ground organisation model for the humanoid robot iCub," Nature communications, vol. 16, no. 1, p. 1874, 2025.

[15] R. Douglas, M. Mahowald, and C. Mead, "Neuromorphic analogue vlsi," Annual review of neuroscience, vol. 18, no. 1, pp. 255–281, 1995.

[16] C. Pehle, S. Billaudelle, B. Cramer, J. Kaiser, K. Schreiber, Y. Stradmann, J. Weis, A. Leibfried, E. Müller, and J. Schemmel, "The brainscales-2 accelerated neuromorphic system with hybrid plasticity," Frontiers in Neuroscience, vol. 16, 2022.

[17] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps)," IEEE transactions on biomedical circuits and systems, vol. 12, no. 1, pp. 106–122, 2017.

[18] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri, "A reconfigurable on-line learning spiking

neuromorphic processor comprising 256 neurons and 128k synapses," Frontiers in neuroscience, vol. 9, p. 141, 2015.

[19] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The spinnaker project," Proceedings of the IEEE, vol. 102, pp. 652–665, 5 2014.

[20] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, et al., "Loihi: A neuromorphic manycore processor with on-chip learning," Ieee Micro, vol. 38, no. 1, pp. 82–99, 2018.

[21] H. B. P. SGA, "Neuromorphic computing," 2020.

[22] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in 2015 International Joint Conference on Neural Networks (IJCNN), pp. 1–8, 2015.

[23] S. B. Shrestha and Q. Song, "Event based weight update for learning infinite spike train," in 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 333–338, 2016.

[24] M. Gehrig, S. B. Shrestha, D. Mouritzen, and D. Scaramuzza, "Event-based angular velocity regression with spiking networks," 2020.

[25] C. M. Parameshwara, S. Li, C. Fermüller, N. J. Sanket, M. S. Evanusa, and Y. Aloimonos, "Spikems: Deep spiking neural network for motion segmentation," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3414–3420, 2021.

[26] M. Fatahi, P. Boulet, and G. D'angelo, "Event-driven nearshore and shoreline coastline detection on spinnaker neuromorphic hardware," Neuromorphic Computing and Engineering, vol. 4, no. 3, p. 034012, 2024.

[27] G. D'Angelo, E. Janotte, T. Schoepe, J. O'Keeffe, M. B. Milde, E. Chicca, and C. Bartolozzi, "Event-based eccentric motion detection exploiting time difference encoding," Frontiers in neuroscience, vol. 14, p. 451, 2020.

[28] G. D'Angelo, A. Perrett, M. Iacono, S. Furber, and C. Bartolozzi, "Event driven bio-inspired attentive system for the iCub humanoid robot on SpiNNaker," Neuromorphic Computing and Engineering, vol. 2, no. 2, p. 024008, 2022.

[29] G. D'Angelo, V. Clerico, C. Bartolozzi, M. Hoffmann, P. M. Furlong, and A. Hadjiivanov, "Wandering around: A bioinspired approach to visual attention through object motion sensitivity," Neuromorphic Computing and Engineering, vol. 5, no. 2, p. 024019, 2025.

[30] G. Rutishauser, M. Scherer, T. Fischer, and L. Benini, "7 μj/inference end-to-end gesture recognition from dynamic vision sensor data using ternarized hybrid convolutional neural networks," Future Generation Computer Systems, vol. 149, pp. 717–731, 2023.

[31] X. Chen, L. Su, J. Zhao, K. Qiu, N. Jiang, and G. Zhai, "Sign language gesture recognition and classification based on event camera with spiking neural networks," Electronics, vol. 12, no. 4, p. 786, 2023.

[32] J. H. Lee, T. Delbruck, M. Pfeiffer, P. K. Park, C.-W. Shin, H. Ryu, and B. C. Kang, "Real-time gesture interface based on event-driven processing from stereo silicon retinas," IEEE transactions on neural networks and learning systems, vol. 25, no. 12, pp. 2250–2263, 2014.

[33] X. Xiao, L. Wang, X. Chen, L. Qu, S. Guo, Y. Wang, and Z. Kang, "Dynamic vision sensor based gesture recognition using liquid state machine," in International Conference on Artificial Neural Networks, pp. 618–629, Springer, 2022.

[34] A. Vasudevan, P. Negri, B. Linares-Barranco, and T. Serrano-Gotarredona, "Introduction and analysis of an event-based sign language dataset," in 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020), pp. 675–682, IEEE, 2020.

[35] E. Ceolini, C. Frenkel, S. B. Shrestha, G. Taverni, L. Khacef, M. Payvand, and E. Donati, "Hand-gesture recognition based on emg and event-based camera sensor fusion: A benchmark in neuromorphic computing," Frontiers in neuroscience, vol. 14, p. 637, 2020.

[36] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, et al., "A low power, fully event-based gesture recognition system," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7243–7252, 2017.

[37] R. Massa, A. Marchisio, M. Martina, and M. Shafique, "An efficient spiking neural network for recognizing gestures with a dvs camera on the loihi neuromorphic processor," in 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–9, IEEE, 2020.

[38] S. Arfa, B. Vogginger, C. Liu, J. Partzsch, M. Schöne, and C. Mayr, "Efficient deployment of spiking neural networks on spinnaker2 for dvs gesture recognition using neuromorphic intermediate representation," in 2025 Neuro Inspired Computational Elements (NICE), pp. 1–8, IEEE, 2025.

[39] K. K. Nazeer, M. Schöne, R. Mukherji, B. Vogginger, C. Mayr, D. Kappel, and A. Subramoney, "Language modeling on a spinnaker2 neuromorphic chip," in 2024 IEEE 6th International Conference on AI Circuits and Systems (AICAS), pp. 492–496, IEEE, 2024.

[40] K. Stewart, G. Orchard, S. B. Shrestha, and E. Neftci, "Online few-shot gesture learning on a neuromorphic processor," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 10, no. 4, pp. 512–521, 2020.

[41] T. Tao, Y. Zhao, T. Liu, and J. Zhu, "Sign language recognition: A comprehensive review of traditional and deep learning approaches, datasets, and challenges," IEEE Access, vol. 12, pp. 75034–75060, 2024.

[42] N. Pugeault and R. Bowden, "Spelling it out: Real-time asl fingerspelling recognition," in 2011 IEEE International conference on computer vision workshops (ICCV workshops), pp. 1114–1119, Ieee, 2011.

[43] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos, "Graph-based spatio-temporal feature learning for neuromorphic vision sensing," IEEE Transactions on Image Processing, vol. 29, pp. 9084–9098, 2020.

[44] "Sign language mnist." https://www.kaggle.com/datasets/datamunge/sign-language-mnist, 2017.

[45] N. Pugeault and R. Bowden, "Spelling it out: Real–time asl fingerspelling recognition," in Proceedings of the IEEE International Conference on Computer Vision, pp. 1114–1119, 2011.

[46] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos, "Graph-based object classification for neuromorphic vision sensing," in Proceedings of the IEEE/CVF international conference on computer vision, pp. 491–501, 2019.

[47] H. Zhou, H. S. Friedman, and R. Von Der Heydt, "Coding of border ownership in monkey visual cortex," Journal of Neuroscience, vol. 20, no. 17, pp. 6594–6611, 2000.

[48] J. K. Eshraghian, M. Ward, E. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, "Training spiking neural networks using lessons from deep learning," Proceedings of the IEEE, vol. 111, no. 9, pp. 1016–1054, 2023.

[49] H. Li, J. Wang, J. Yuan, Y. Li, W. Weng, Y. Peng, Y. Zhang, Z. Xiong, and X. Sun, "Event-assisted low-light video object segmentation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3250–3259, 2024.

[50] M. Mohammadi, P. Chandarana, J. Seekings, S. Hendrix, and R. Zand, "Static hand gesture recognition for american sign language using neuromorphic hardware," Neuromorphic Computing and Engineering, vol. 2, no. 4, p. 044005, 2022.

[51] "Icns event based camera simulator." https://github.com/neuromorphicsystems/IEBCS, 2021.

[52] S. J. van Albada, A. G. Rowley, J. Senk, M. Hopkins, M. Schmidt, A. B. Stokes, D. R. Lester, M. Diesmann, and S. B. Furber, "Performance comparison of the digital neuromorphic hardware spinnaker and the neural network simulation software nest for a full-scale cortical microcircuit model," Frontiers in Neuroscience, vol. Volume 12 - 2018, 2018.

[53] E. Stromatias, F. Galluppi, C. Patterson, and S. Furber, "Power analysis of large-scale, real-time neural networks on spinnaker," in The 2013 international joint conference on neural networks (IJCNN), pp. 1–8, IEEE, 2013.

[54] E. Stromatias, D. Neil, F. Galluppi, M. Pfeiffer, S.-C. Liu, and S. Furber, "Scalable energy-efficient, low-latency implementations of trained spiking deep belief networks on spinnaker," in 2015 International Joint Conference on Neural Networks (IJCNN), pp. 1–8, IEEE, 2015.

[55] M. Rivera-Acosta, S. Ortega-Cisneros, J. Rivera, and F. Sandoval-Ibarra, "American sign language alphabet recognition using a neuromorphic sensor and an artificial neural network," Sensors, vol. 17, no. 10, p. 2176, 2017.