2020 8th IEEE International Conference on Biomedical
Robotics and Biomechatronics (BioRob)
New York, USA. Nov 29 - Dec 1, 2020

Embodied Neuromorphic Vision with Continuous Random Backpropagation

Jacques Kaiser[1], Alexander Friedrich[1], J. Camilo Vasquez Tieck[1],
Daniel Reichard[1], Arne Roennau[1], Emre Neftci[2], Rüdiger Dillmann[1]

***Abstract*— The brain outperforms computer architectures in aspects of energy efficiency, robustness and adaptivity. Brain computations are modeled in silico with spiking neural networks and neuromorphic hardware. Recently, three-factor synaptic plasticity rules approximating backpropagation have been derived. Suited to neuromorphic hardware, these rules can learn online with asynchronous updates. In this paper, we present Continuous Random Backpropagation (cRBP), a continuous version of Event-Driven Random Backpropagation. This learning rule performs comparably to state-of-the-art rules on the DvsGesture dataset. We additionally show that the accuracy can be significantly increased with a simple attention mechanism. This mechanism provides translation invariance at low computational cost compared to convolutions by exploiting event stream sparsity. Subsequently, we integrate cRBP in a real robotic setup, where a gripper grasps objects according to the detected visual affordances. In this setup, visual information is actively sensed by a Dynamic Vision Sensor (DVS) mounted on a robotic head performing microsaccadic eye movements. Our results suggest that advances in neuromorphic technology and plasticity rules enable the development of learning robots operating at high speed and low power.**

I. INTRODUCTION

The brain outperforms computer architectures in aspects of energy efficiency, robustness and adaptivity. The computational paradigms of the brain are vastly different from modern computer architectures. Biological neural networks base their computations on local information and communicate asynchronously with spikes. Understanding how these paradigms can be implemented in hardware would enable the design of autonomous learning robots operating at high speed for a fraction of the energy budget of current solutions.

Learning in the brain is believed to be based on synaptic plasticity. Unlike conventional machine learning methods, synaptic plasticity rules characterize weight updates in terms of information local to the synapse. Synaptic learning enables an efficient neuromorphic hardware implementation, asynchronous updates and online learning.

Recently, a family of synaptic plasticity rules for training multi-layer spiking neural networks have been proposed in [1]–[4]. These rules implement variations of backpropagation by approximating gradients as a multiplication of three factors related to the input, output and error of a synapse [5]. In this paper, we evaluate the ability to efficiently learn spatio-temporal visual representations using three-factor rules embodied in a robotic setup.

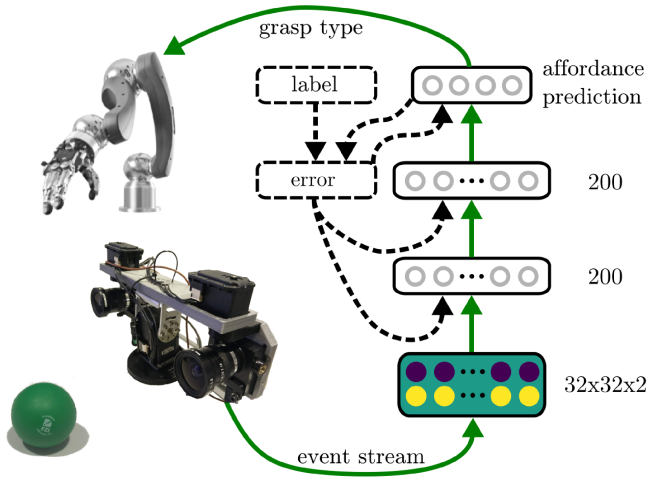


Fig. 1: Our robotic setup to embody the synaptic learning rule cRBP. The DVS is mounted on a robotic head performing microsaccadic eye movements. The spiking network is trained online (dashed-line connections) in a supervised fashion to classify visual affordances from the event streams. The output neurons of the network correspond to the four types of affordances: ball-grasp, bottle-grasp, pen-grasp or do nothing. At test time, a Schunk LWA4P arm equipped with a Schunk SVH 5-finger hand performs the corresponding reaching and grasping motion.

We present the Continuous Random Backpropagation (cRBP) rule – a continuous version of Event-Driven Random Backpropagation (eRBP) [1] – following the derivation of Deep Continuous Local Learning (DECOLLE) [3]. Like DECOLLE, the synaptic weights have continuous-time dynamics, unlike eRBP which only updates synaptic weights on pre-synaptic spikes. Like eRBP, the error signals for the hidden neurons are computed at the network output, unlike DECOLLE which computes errors locally at the layer. These rules learn in an online fashion, in the sense that synaptic weights are updated while the input is streamed in the network, by propagating information required to compute the gradient forward as Real-Time Recurrent Learning (RTRL) [6]. This enables the space complexity of these rules to remain constant with respect to time. In contrast, other rules such as SLAYER [7] based on Backpropagation-Through-Time (BPTT) require to store an history of the past neural activity. In this case, memory consumption increases with

[1]FZI Research Center For Information Technology, Karlsruhe, Germany {jkaiser,friedric,tieck,daniel.reichard, roennau,dillmann}@fzi.de
[2]Department of Cognitive Sciences, University of California Irvine, Irvine, USA neftci@uci.edu

the length of the time sequence, an important limitation, as reported in [8]. On the other hand, DECOLLE and cRBP can learn spatio-temporal patterns on long sequences with a fine temporal resolution. This makes them perfectly suited to neuromorphic vision sensor data. Additionally, unlike conventional artificial neural networks which require to integrate frames from event streams [9]–[11], our method is suitable to low latency applications.

We show that the accuracy of cRBP is comparable to state-of-the-art methods on the IBM DvsGesture dataset [12]. A covert attention mechanism is introduced which further improves the efficiency and accuracy of the learning rules by providing translation invariance at low computational cost compared to convolutions. Inspired by receptive field remapping in the visual cortex, this attention mechanism is tailored to the sparsity of the visual event streams. Finally, we integrate cRBP in a real-world closed-loop robotic grasping setup involving a robotic head, arm and a 5-finger hand. The spiking network learns to classify different types of affordances based on visual information obtained with microsaccades, and communicates this information to the arm for grasping. This real-world task has the potential to enhance neuromorphic and neurorobotics research since many functional components such as reaching, grasping and depth perception can be easily segregated and implemented with brain models. This work paves the way towards the integration of brain-inspired computational paradigms into the field of robotics.

A barrier to embodied learning robots is the offline (batch learning) nature of conventional implementations of backpropagation. In comparison, our model can learn from events streamed from the DVS [26] with little loss in accuracy. This enables continual updates without separation of training and testing phases. However, such life-long learning setups require to address the forgetting problem resulting from learning on temporally correlated input data.

II. METHOD

A. Continuous Random Backpropagation

The backpropagation algorithm computes the gradient of a synaptic weight with respect to an arbitrary loss function defined on the network's output. The credit of a neuron – how a change in its output affects the loss – therefore depends on the synaptic weights of the subsequent layers. This weight transport problem is solved by Direct Feedback Alignment [13], an instance of random backpropagation [14], by computing the credit for a neuron i as a linear combination of network errors e_k with fixed, random coefficients g_{ik}. This solution also enables asynchronous weight updates by decoupling the conventional forward and backward phases of backpropagation. An adaptation of Direct Feedback Alignment to spiking networks of Leaky Integrate-And-Fire neurons was derived in [1]. Named eRBP, this synaptic plasticity rule can be formulated as:

$$\Delta w_{ij}(t) \propto \sum_{k \in rdout} e_k(t) g_{ik} \times \Theta'(u_i(t)) \times s_j(t), \quad (1)$$

with w_{ij} the synaptic weight from j to i, Θ' the derivative of the spike function, u_i the membrane potential of neuron i and s_j the pre-synaptic spiketrain (either 0 or 1 at a given time t). The set $rdout$ contains the indices of the readout neurons y_k. The spike function Θ is the non-differentiable heaviside function (hard threshold), but its derivative can be approximated with a surrogate gradient [15]. As with eRBP and DECOLLE, we approximate this derivative with the boxcar function: $\Theta'(x) \approx Boxcar(x) = 1$ if $-0.5 < x < 0.5$, otherwise 0. The symbol $\propto$ refers to a proportionality relation – the multiplicative constant is the learning rate which can be chosen freely.

In the special case of a Mean Square Error (MSE) loss, the errors e_k are computed as the difference between network readouts y_k and network targets $\hat{y_k}$:

$$\begin{aligned} e_k(t) &= y_k(t) - \hat{y_k}(t), \\ y_k(t) &= \sum_{i \in out} s_i(t) \times g_{ik} \end{aligned} \quad (2)$$

with out the set containing the indices of the output neurons.

Since the weight update in Equation (1) is proportional to s_j, weight updates in eRBP are triggered by pre-synaptic spikes, in an event-driven fashion. However, this formulation does not account for the dynamics of the post-synaptic potentials. More recent three-factor rule derivations now incorporate an eligibility trace to account for this dynamics [2]–[4] (see Equation (4) in [2]). We can integrate this term directly into Equation (1), yielding the cRBP rule:

$$\Delta w_{ij}(t) \propto \sum_{k \in rdout} e_k(t) g_{ik} \times \Theta'(u_i(t)) \times \epsilon * s_j(t), \quad (3)$$

where $*$ denotes a temporal convolution and ϵ is the post-synaptic potential kernel. This new rule describes continuous synapse dynamics rather than event-driven updates – we therefore refer to it as cRBP. The main difference with SuperSpike is the loss function: SuperSpike relies on a van Rossum distance with a target spiketrain. This leads SuperSpike to require one eligibility trace per synapse, whereas cRBP requires only one eligibility trace per neuron. Additionally, the computation of this eligibility trace can be factored into the neural dynamics, as presented in DECOLLE. The main difference with DECOLLE is that DECOLLE relies on local readouts y_k^l and local targets $\hat{y}_k^l$ for every layer l to compute the errors e_k^l. Instead, hidden layers in cRBP are updated with respect to the global network loss.

The simulations presented in this paper rely on the same neuron model as DECOLLE, introduced in Equation 4 in [3]. Note that this neuron model does not account for synaptic delays, and the refractory period is approximated with a self-inhibition.

B. Network Architecture

The network is presented in Figure 1. It learns from event streams provided by a DVS. Since spikes are not signed events, we associate two neurons for each pixel to convey ON- and OFF-events separately. This distinction is important since event polarities carry information about the

direction of motion (see Figure 3). Since events are emitted only upon light change, two different setups are analyzed: a dataset where changes originate from motion in the scene, and a dataset where changes originate from fixational eye movements. The evaluation on these two types of dataset can lead to different performance [16].

Only spikes are propagated from a layer to another. However, the errors computed at the network output (e_k in Equation (2)) are communicated to the layers as an analog value. A previous implementation of the work presented in this paper relied on eRBP which was implemented with Auryn [17][1]. This implementation computed and communicated errors using only neural dynamics and spikes, as in [1]. The newer implementation of this work is based on PyTorch which offers more tools for learning such as auto-differentiation, convolution, max pooling and advanced optimization methods.

C. Covert Attention Window

It was shown in biology that receptive fields of frontal eye field neurons are constantly remapped [18]–[20]. Inspired from this insight, we introduce a simple covert attention mechanism which consists of continuously moving an attention window across the input stream when new events are received. Covert attention, as opposed to overt attention, signifies an attention shift which was not marked by eye movements. Particularly suited to the sparsity of event streams, the center of the attention window is computed online as the median address of the last $n_{\text{attention}}$ events, see Figure 3. By remapping receptive fields relatively to the center of the motion, this technique enables translation invariance at low computational cost compared to convolutions. Indeed, convolutions process all the regions of the image identically and require a weight sharing mechanisms complicated to implement on neuromorphic hardware. Our method also allows to reduce the dimension of the event stream without rescaling, thus decreasing the size of the neural network.

A similar method was already introduced in [21] for classifying a dataset of three human motions (bend, sit/stand, walk) recorded with a DVS. Their approach consists of remapping the address of their feature neurons (C1) with respect to their mean activation before being fed to the classifier. Instead, our method consists of remapping the address events directly, with respect to the median address of the last events. Unlike the median, the mean activation can result in an event-less attention window in case of multiple objects in motion, such as two-hand gestures. Additionally, since our attention window is smaller than the event stream, eccentric events are not processed by the network. We show in this paper how this biologically motivated technique boosts the performance, even on DvsGesture, where multiple body parts are simultaneously in motion. We note that a similar mechanism could be integrated in a robotic head as the one used in this paper to perform saccadic eye movements (see Figure 2). In this case, an additional mechanism to discard events resulting of the ego-motion would be required.

[1] https://github.com/HBPNeurorobotics/auryn

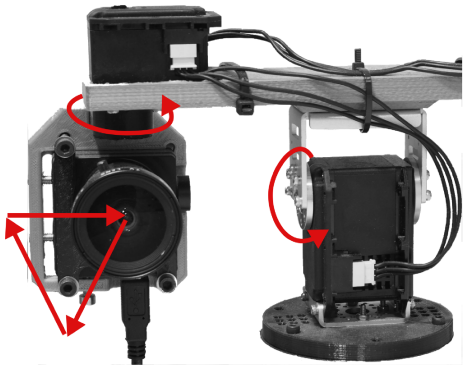


Fig. 2: Microsaccadic motion of the DVS performed by the robotic head.

D. Microsaccadic eye-movements

For our real-world grasping experiment, address events are sensed from static scenes by performing microsaccadic eye movements. This technique was already used to convert images to event streams [22], essentially extracting edge features [16]. To this end, we mounted the DVS on the robotic head presented in [23], see Figure 2. One Dynamixel servo MX-64AT is used to tilt both DVS simultaneously, while two other Dynamixel servos MX-28AT are used to pan each DVS independently. The center of all rotations is approximately the optical center of each DVS. In this work, only the events of the right DVS are processed. The microsaccadic motion consists of an isosceles triangle in joint space, with each motion lasting $0.2\,\text{s}$. The motions are a negative tilt of α and negative pan of $\alpha/2$, followed by a tilt of α and negative pan of $\alpha/2$, finalized with a return to the initial position. We chose the angle $\alpha = 1.833°$. This angle is much smaller in biology, but DVS pixels are much larger than the photoreceptors of the retina [24]. The precise microsaccadic motion is not relevant for learning, but similar motions should be used for training and testing.

The microsaccades are triggered either manually for recording training data, or automatically in a loop at test time. We allow the events to flow through the network only when a microsaccade is triggered. No information about the properties of the microsaccade is passed as an input to the network.

III. EVALUATION

Two different network architectures are used in this work: a convolutional network and a dense network. Both architectures expect input dimensions 2x32x32 at a given time step. The first architecture is the same 3-layers convolutional network used in DECOLLE [3]. The convolutions consist of 64, 128 and 128 kernels of size 7x7 respectively, interleaved with max pooling and spike dropout operations. The max pooling operation is applied before the spike function Θ. The dense network consists of two hidden layers with 200

Model	Accuracy	#Iterations
IBM EEDN (conv) [12]	91.77%(94.59%)	Offline 64M
SLAYER (conv) [7]	93.64 ± 0.49 %	Offline .27M
DECOLLE (conv) [3]	95.60 ± 0.56%	Online .16M
DECOLLE (conv+attention)	$\mathbf{96.37 \pm 0.51}$%	**Online .16M**
cRBP (conv)	92.48 ± 0.89%	Online .16M
cRBP (conv+attention)	95.34 ± 0.78%	Online .16M
cRBP (2L dense)	77.93 ± 2.09%	Online .16M
cRBP (2L dense+attention)	90.80 ± 1.16%	Online .16M

TABLE I: Classification accuracy on the DvsGesture dataset. The mean accuracies and standard deviations reported in this work (below the horizontal bar) were computed over 7 runs. The number of iterations refers to the number of training samples that were fed to the network. EEDN increases its accuracy with output filtering.

neurons each. The implementation for cRBP is realized with the PyTorch framework and integrated in the open-source DECOLLE code base[2]. This allows a fair comparison between DECOLLE and cRBP, using the same training procedure and neural parameters, see [3] for details. Specifically, the loss function L is a smooth L1 loss, the learning rate is set to 10^{-9} and divided by 5 every 30 epochs. The optimizer is AdaMax [25] with parameters $\beta_1 = 0$, $\beta_2 = 95$.

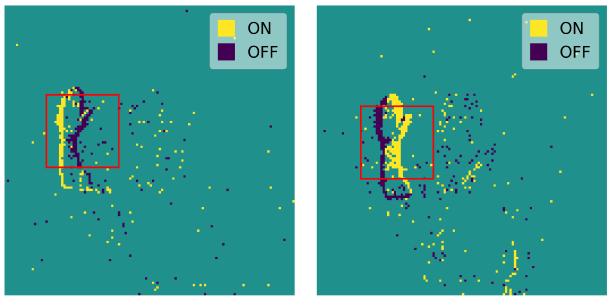
A. DvsGesture

We evaluate cRBP on the DvsGesture dataset, following the same training procedure as the DECOLLE rule [3], currently achieving state-of-the-art accuracy on this dataset. DvsGesture is an action recognition dataset recorded by IBM using a DVS [12], [26]. It consists of 1342 recordings of 29 subjects performing 11 diverse actions in three different illumination conditions. This dataset is loaded into PyTorch using the torchneuromorphic library[3] developed in [3]. Specifically, training samples consist of 500 ms-long event streams, and test samples 1800 ms-long. These samples are sliced at random location in the dataset, but ensuring that the motion is presented during the whole sequence. This procedure maximizes the use of the dataset, leading to 1176 train samples and 288 test samples. The sequences were presented to the network in mini-batches of 72 samples.

The event streams recorded from the DVS are 2-channels (ON and OFF events) with 128x128 pixels. We compare the accuracy of the network on the downsized streams and using the covert attention window mechanism described in Section II-C. The downsize operation reduced the event stream to 32x32 by grouping neighboring pixels, and was used in [3]. The attention window re-address the events in a 32x32 window with respect to the median event. It was implemented as an alternative to the downsize operation in the torchneuromorphic library. The number of events to calculate the position of the attention window was set to $n_{\text{attention}} = 1000$.

The final accuracies for the different experiments on DvsGesture are reported in Table I. Our evaluation on DvsGesture shows that cRBP efficiently learns spatio-temporal patterns to classify motions from raw event streams. With the same convolutional architecture and training procedure as DECOLLE, cRBP reaches 92.48% accuracy, close to state-of-the art accuracy, see Figure 4. When replacing the downsampling operation of the event stream with the attention mechanism, the accuracy further increases to 95.34% for cRBP and to 96.37% for DECOLLE. This improvement is more significant for the dense network architecture. In this case, the attention window mechanism leads to a substantial improvement from 77.93% to 90.80% accuracy compared to the downsampling approach. This confirms our assumption that the attention window mechanism provides translation invariance with respect to the performed gestures. The reason why the performance of the convolutional networks only slightly improves is because convolution and max pooling operations already provide translation invariance. It results from the same kernel being convolved on the whole image

[2]https://github.com/nmi-lab/decolle-public
[3]https://github.com/nmi-lab/torchneuromorphic



(a) Arm circling clockwise (b) Arm circling anticlockwise

Fig. 3: Aggregation of 1000 events for two samples of the DvsGesture dataset (user 10). The information about direction of motion is contained in the event polarity, hence the importance of their segregation in the input layer. The red square represents the attention window of size 32x32, calculated as the median address of the last 1000 events.

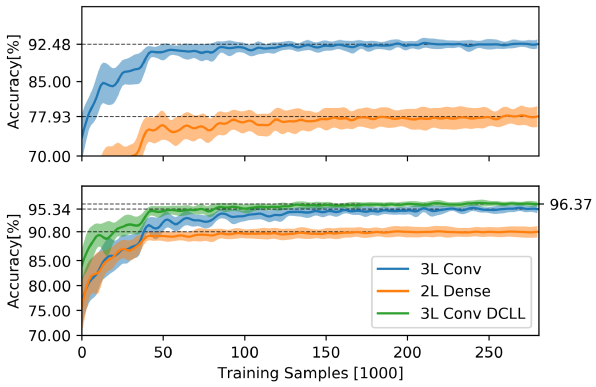


Fig. 4: Classification accuracy on DvsGesture with the convolutional and dense network architectures on the 32x32 event streams. Top: downsampling. Bottom: attention window mechanism. Shaded areas represent the standard deviation computed over 7 runs with different random seeds.

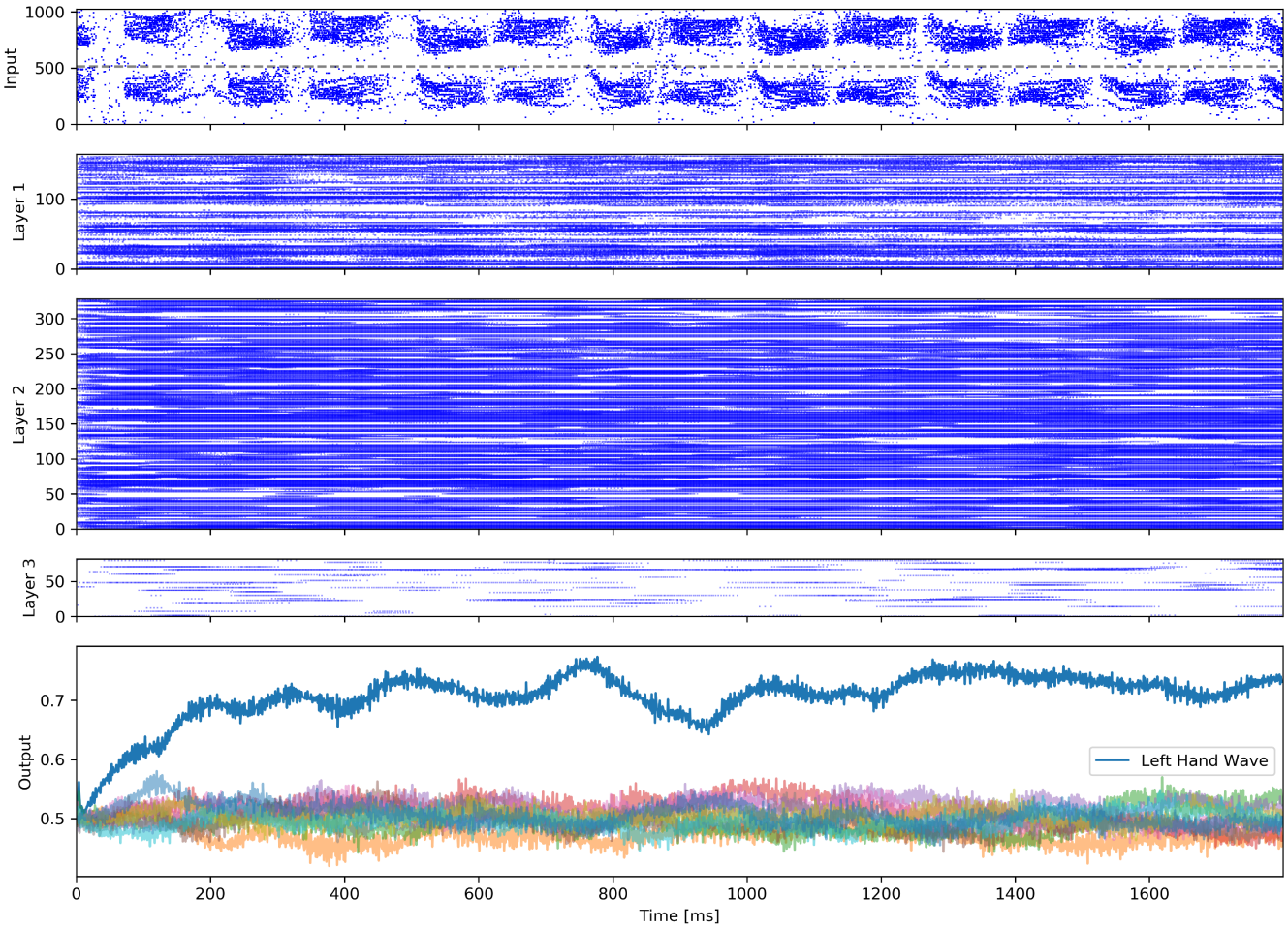


Fig. 5: Spiketrains of cRBP with attention window for a test sample of class "Left Hand Wave" from the DvsGesture dataset (dropout deactivated). The spiketrains only show 50% of the input neurons and 1% of the hidden neurons for readability. The rhythm of the "Left Hand Wave" motion is clearly visible in the input spiketrain. Lack of regularization in the loss and weak refractory term in the neuron dynamics lead to pathologically high spike rates. Readout activities are filtered with a sigmoid function.

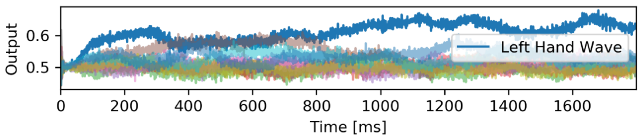


Fig. 6: Output of cRBP with downsampling for the same test sample as shown in Figure 5.

to form a feature map. We therefore expect that locally connected layers (convolutional topology where kernels are not shared for the whole the image [27]) coupled with the presented attention mechanism could drastically reduce the amount of computations while retaining the accuracy of a convolutional network. Such networks are also more biologically plausible than convolutional networks since no mechanisms in the brain is known to support weight sharing.

The improvement of the attention mechanism over downsampling is also reflected in the classification output of a test sample. Indeed, with the attention mechanism, the network unambiguously and correctly classifies the test sample early in the sequence, see Figure 5. With the downsampling approach, the confusion in the output of the network is higher, see Figure 6. We note that many neurons in the hidden layers spike with very high rates, including the maximum rate of 1000Hz imposed by the simulation time step of 1ms (neglecting dropout). Indeed, the weak refractory term in the neural dynamics decreases the membrane potential after a spike, but does not prevent subsequent spikes. A lower spiking rate can be favored by adding a regularization term in the loss function as mentioned in [3]. This is shown in the grasping experiment, see Figure 9.

B. Grasp-type Recognition

In this experiment, we embody cRBP in the real-world grasping robotic setup shown in Figure 7. In this setup, the spiking network is trained to recognize four labels corre-

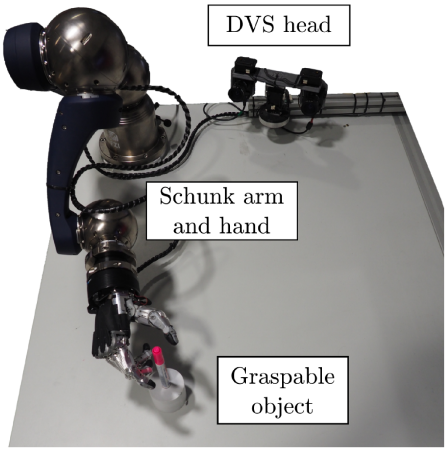


Fig. 7: Real-world grasp-type Recognition experiment setup integrating a Schunk LWA4P arm equipped with a Schunk SVH 5-finger hand and a DVS head. The DVS head performs microsaccadic eye movements to sense event streams from static scenes. We recorded a small four-classes dataset (ball, bottle, pen, background) of 50 samples per class. At test time, the detected grasp-type triggers the corresponding predefined reaching and grasping motion.

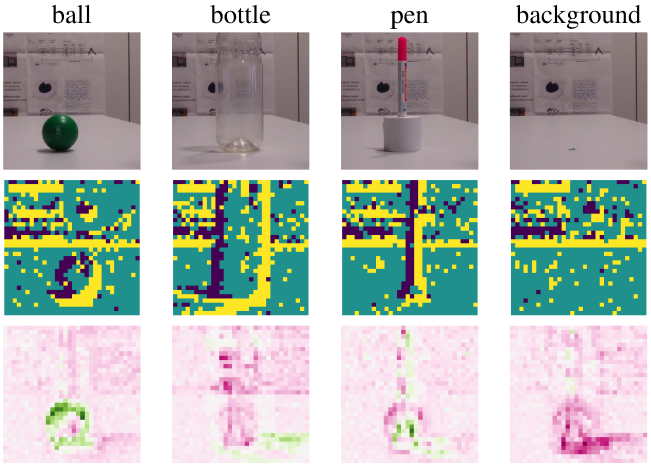


Fig. 8: Example samples and learned weights for the grasp-type recognition experiment. Top row: camera image of the objects. Middle row: integration of the address events after microsaccade onset. Bottom row: projection of the synaptic weights for each label neuron onto the input after training. Green denotes excitation and pink denotes inhibition.

sponding to four different grasps: ball-grasp, bottle-grasp, pen-grasp or do nothing [28]. During training, an object of a particular class is placed on a table at a specific position. The robotic head performs microsaccadic eye movements (similar to the N-MNIST dataset [22]) to extract visual information from the static object. Only the event stream of one DVS is recorded, together with the corresponding object affordance. In this experiment, the attention window of dimension 32x32 is fixed to match the position of the objects on the table, see Figure 8 for example samples. During testing, a microsaccade is performed and the detected object affordance triggers the adequate predefined reaching and grasping motion on a Schunk LWA4P arm equipped with a Schunk SVH 5-finger hand. This demonstrator was implemented with the ROS Framework [29] and the ROS DVS driver introduced in [30].

With only 50 samples per class and 10 epochs, the network was capable of learning the four visual affordances (see the supplementary video[4] using the previous eRBP implementation). Example spiketrains and classification results at test time are shown in Figure 9. Spike rates are kept lower than in the DvsGesture experiment by using regularization in the loss function. Specifically, the loss function becomes:

$$\mathcal{L} = L + \sum_l \lambda_1 \langle [U_i^l + 0.01]^+ \rangle_i + \lambda_2 [0.1 - \langle U_i^l \rangle_i]^+ \quad (4)$$

where L is the network loss, U_i^l is the membrane potential of neuron i in layer l, the $\langle \cdot \rangle_i$ denotes averaging over index i, $[\cdot]^+$ is a linear rectification, $\lambda_1 = 2.5 \cdot 10^{-2}$ and $\lambda_2 = 1.5 \cdot 10^{-4}$ for both layers. The term with the λ_1 factor favors a minimum firing rate, and the term with the λ_2 factor keeps the membrane potential below threshold on average. This regularization decreases the average spiking rate although we note that individual neurons can still spike with high rates up to 500Hz (neglecting dropout) for the second layer, see Figure 9.

The network readout for the correct class is high ($> .66$) shortly after microsaccade onset: 43ms for the ball, 58ms for the bottle, 35ms for the pen and 33ms for the background, see Figure 9. These numbers are coherent with behavioral experiments on humans quantifying the reaction time to a visual stimuli [31], [32]. This resemblance should be further investigated on tasks identical to those used in the behavioral experiments. To this end, different neural dynamics enforcing plausible spike rates and including synaptic delays should be used.

Since the DVS does not sense colors, the network only relies on shape information, crucial for affordances. This allowed the network to moderately generalize despite the small amount of training samples. The learned weights projected to the input are displayed in Figure 9. A single object per affordance was used during training, but the network could recognize objects with different colors of the same shape. Recognition also worked when the objects were slightly moved from the reference point used for grasping. However, the network was not robust to change in background or unexpected background motions happening during the microsaccade. This is due to the background being learned as an additional class for the "do nothing" affordance.

IV. CONCLUSION

Neuromorphic engineering technology enables the design of autonomous learning robots operating at high speed for a fraction of the energy consumption of current solutions. Until

[4]https://neurorobotics-files.net/index.php/s/sBQzWFrBPoH9Dx7

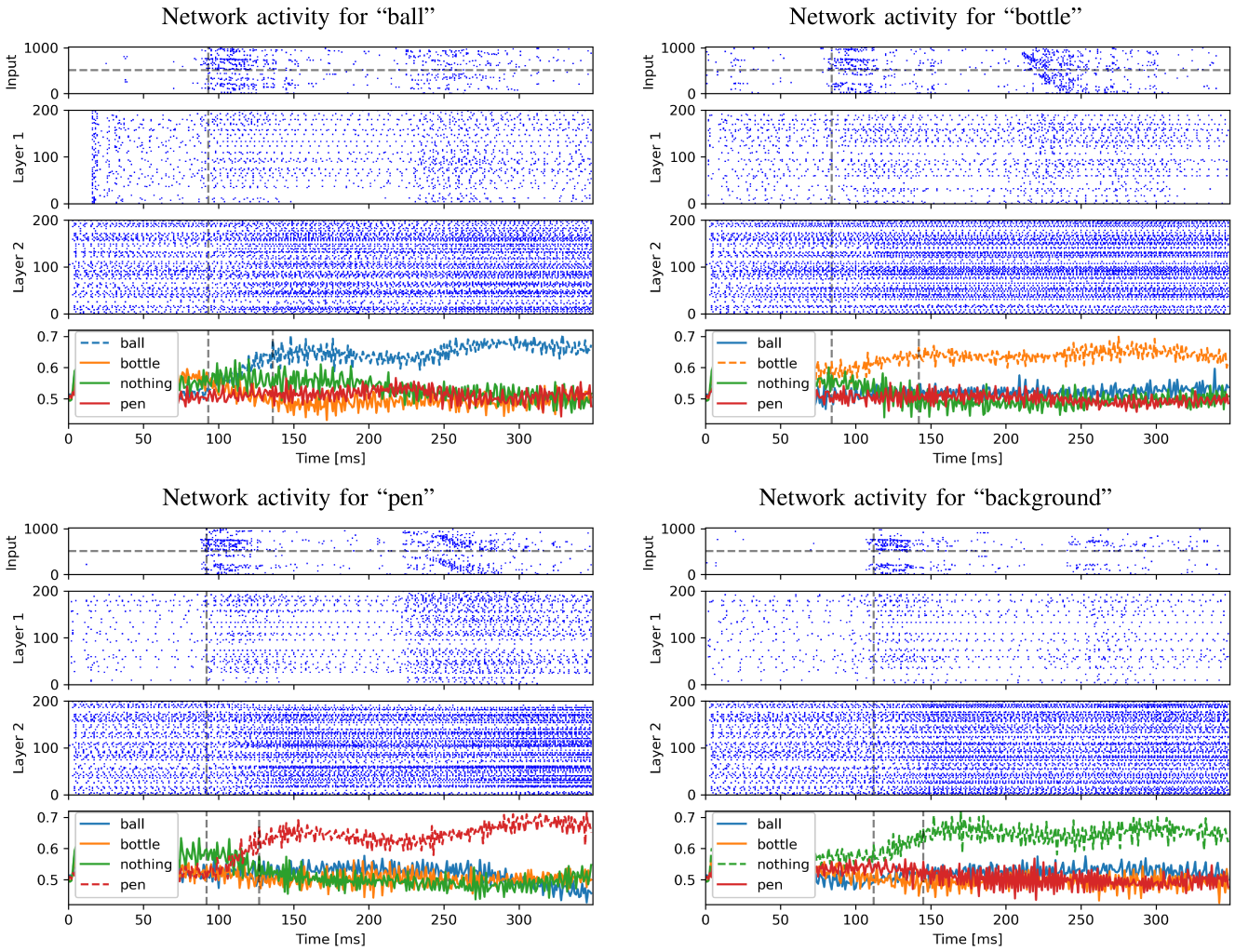


Fig. 9: Spiketrains and classification results for four test samples of our grasp-type dataset. The network manages to correctly classify the four test samples. The vertical lines denote the microsaccade onset and high detection confidence (readout > 0.66) respectively. The phases of the microsaccadic motion are visible in the input spiketrains (first row of each plot), see Section II-D.

recently, the advantages of this technology were limited due to the lack of synaptic plasticity rules for training multi-layer spiking networks. This bottleneck has been addressed since the derivation of three-factor rules approximating back-propagation. In this paper, we demonstrated the ability of cRBP to learn spatio-temporal representations from event streams provided by a DVS. With the addition of a simple biologically-inspired covert attention mechanism, we have shown that cRBP and DECOLLE further improved their accuracy on the DvsGesture benchmark in comparison to classical rescaling approaches. This attention mechanism provides translation invariance at a low computational cost compared to convolutions. Lastly, we integrated cRBP in a real-world robotic grasping experiment, where affordances are detected from microsaccadic eye movements and conveyed to a robotic arm and hand setup for execution. Real robot learning experiments are challenging because of the difficulty and time required to collect relevant training data. Our results show that correct affordances are detected within about 40ms after microsaccade onset, which is coherent with biological findings in humans. For future work, these results should be further investigated by replicating the behavioral experiments presented in [31], [32]. Additionally, other components of the grasp-type recognition experiment could be implemented with spiking networks, such as reaching motions [33], [34], grasping motions [35] and depth perception [23]. It was already shown in [8] that spiking networks can learn regression tasks from event streams. This would enable a wider variety of computational brain models to be compared against behavioral experimental results in real-world scenarios. This work paves the way towards the integration of brain-inspired computational paradigms into the field of robotics.

ACKNOWLEDGMENT

This research has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 785907 (Human Brain Project SGA2).

REFERENCES

[1] E. O. Neftci, C. Augustine, S. Paul, and G. Detorakis, "Event-driven random back-propagation: Enabling neuromorphic deep learning machines," *Frontiers in neuroscience*, vol. 11, p. 324, 2017.

[2] F. Zenke and S. Ganguli, "Superspike: Supervised learning in multilayer spiking neural networks," *arXiv preprint arXiv:1705.11146*, 2017.

[3] J. Kaiser, H. Mostafa, and E. Neftci, "Synaptic plasticity dynamics for deep continuous local learning (decolle)," *Frontiers in Neuroscience*, vol. 14, p. 424, 2020. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2020.00424

[4] G. Bellec, F. Scherr, E. Hajek, D. Salaj, R. Legenstein, and W. Maass, "Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets," *arXiv preprint arXiv:1901.09049*, 2019.

[5] J.-P. Pfister, T. Toyoizumi, D. Barber, and W. Gerstner, "Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning," *Neural computation*, vol. 18, no. 6, pp. 1318–1348, 2006.

[6] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[7] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," in *Advances in Neural Information Processing Systems*, 2018, pp. 1412–1421.

[8] M. Gehrig, S. B. Shrestha, D. Mouritzen, and D. Scaramuzza, "Event-based angular velocity regression with spiking networks," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, p. forthcoming.

[9] D. Gehrig, A. Loquercio, K. G. Derpanis, and D. Scaramuzza, "End-to-end learning of representations for asynchronous event-based data," *arXiv preprint arXiv:1904.08245*, 2019.

[10] G. Gallego, H. Rebecq, and D. Scaramuzza, "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3867–3876.

[11] G. Gallego, M. Gehrig, and D. Scaramuzza, "Focus is all you need: Loss functions for event-based vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 280–12 289.

[12] 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*, 2017, pp. 7243–7252.

[13] A. Nøkland, "Direct feedback alignment provides learning in deep neural networks," in *Advances in neural information processing systems*, 2016, pp. 1037–1045.

[14] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, "Random synaptic feedback weights support error backpropagation for deep learning," *Nature communications*, vol. 7, p. 13276, 2016.

[15] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks," *arXiv preprint arXiv:1901.09948*, 2019.

[16] J. Kaiser, G. Lindner, J. C. V. Tieck, M. Schulze, M. Hoff, A. Roennau, and R. Dillmann, "Microsaccades for asynchronous feature extraction with spiking networks," in *International Conference on Development and Learning and Epigenetic Robotics (ICDL-EPIROB)*, 2018.

[17] F. Zenke and W. Gerstner, "Limits to high-speed simulations of spiking neural networks using general-purpose computers," *Frontiers in neuroinformatics*, vol. 8, p. 76, 2014.

[18] M. Zirnsak, N. A. Steinmetz, B. Noudoost, K. Z. Xu, and T. Moore, "Visual space is compressed in prefrontal cortex before eye movements," *Nature*, vol. 507, no. 7493, p. 504, 2014.

[19] T. B. Crapse and M. A. Sommer, "Frontal eye field neurons with spatial representations predicted by their subcortical input," *Journal of Neuroscience*, vol. 29, no. 16, pp. 5308–5318, 2009.

[20] M. A. Sommer and R. H. Wurtz, "Influence of the thalamus on spatial visual processing in frontal cortex," *Nature*, vol. 444, no. 7117, p. 374, 2006.

[21] B. Zhao, R. Ding, S. Chen, B. Linares-Barranco, and H. Tang, "Feedforward Categorization on AER Motion Events Using Cortex-Like Features in a Spiking Neural Network." *IEEE transactions on neural networks and learning systems*, vol. PP, no. 99, p. 1, 2014.

[22] G. Orchard, A. Jayawant, G. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *arXiv preprint arXiv:1507.07629*, 2015.

[23] J. Kaiser, J. Weinland, P. Keller, L. Steffen, J. C. V. Tieck, D. Reichard, A. Roennau, J. Conradt, and R. Dillmann, "Microsaccades for neuromorphic stereo vision," in *International Conference on Artificial Neural Networks*. Springer, 2018, pp. 244–252.

[24] S. Martinez-Conde, S. L. Macknik, and D. H. Hubel, "The role of fixational eye movements in visual perception," *Nature Reviews Neuroscience*, vol. 5, no. 3, pp. 229–240, 2004.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128x128 120 db 15us latency asynchronous temporal contrast vision sensor," *IEEE journal of solid-state circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[27] K. Gregor and Y. LeCun, "Emergence of complex-like cells in a temporal product network with local receptive fields," *arXiv preprint arXiv:1006.0448*, 2010.

[28] J. Kaiser, D. Zimmerer, J. C. V. Tieck, S. Ulbrich, A. Roennau, and R. Dillmann, "Spiking convolutional deep belief networks," in *International Conference on Artificial Neural Networks*. Springer, 2017, pp. 3–11.

[29] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[30] E. Mueggler, B. Huber, and D. Scaramuzza, "Event-based, 6-dof pose tracking for high-speed maneuvers," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2761–2768.

[31] S. M. Crouzet, H. Kirchner, and S. J. Thorpe, "Fast saccades toward faces: face detection in just 100 ms," *Journal of vision*, vol. 10, no. 4, pp. 16–16, 2010.

[32] J. G. Martin, C. E. Davis, M. Riesenhuber, and S. J. Thorpe, "Zapping 500 faces in less than 100 seconds: Evidence for extremely fast and sustained continuous visual search," *Scientific reports*, vol. 8, no. 1, p. 12482, 2018.

[33] J. C. V. Tieck, L. Steffen, J. Kaiser, D. Reichard, A. Roennau, and R. Dillmann, "Combining motor primitives for perception driven target reaching with spiking neurons," *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, vol. 13, no. 1, pp. 1–12, 2019.

[34] J. C. V. Tieck, L. Steffen, J. Kaiser, A. Roennau, and R. Dillmann, "Controlling a robot arm for target reaching without planning using spiking neurons," in *2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*. IEEE, 2018, pp. 111–116.

[35] J. C. V. Tieck, H. Donat, J. Kaiser, I. Peric, S. Ulbrich, A. Roennau, M. Zöllner, and R. Dillmann, "Towards grasping with spiking neural networks for anthropomorphic robot hands," in *International Conference on Artificial Neural Networks*. Springer, 2017, pp. 43–51.