Neuromorphic Optical Flow and Real-time Implementation with Event Cameras

Yannick Schnider[1,2], Stanisław Woźniak[1], Mathias Gehrig[3], Jules Lecomte[4], Axel von Arnim[4],
Luca Benini[2,5], Davide Scaramuzza[3], Angeliki Pantazi[1]
[1]IBM Research – Zurich [2]ETH Zurich [3]University of Zurich [4]fortiss GmbH [5]Università di Bologna

Abstract

Optical flow provides information on relative motion that is an important component in many computer vision pipelines. Neural networks provide high accuracy optical flow, yet their complexity is often prohibitive for application at the edge or in robots, where efficiency and latency play crucial role. To address this challenge, we build on the latest developments in event-based vision and spiking neural networks. We propose a new network architecture, inspired by Timelens, that improves the state-of-the-art self-supervised optical flow accuracy when operated both in spiking and non-spiking mode. To implement a real-time pipeline with a physical event camera, we propose a methodology for principled model simplification based on activity and latency analysis. We demonstrate high speed optical flow prediction with almost two orders of magnitude reduced complexity while maintaining the accuracy, opening the path for real-time deployments.

1. Introduction

Optical flow is defined as an apparent motion of objects, edges and surfaces in a visual scene registered by the observer. It is caused by the relative motion between the observer and the scene and does not distinguish between actual motion in the visual scene and change in the observer's pose. The applications of optical flow in the field of computer science include motion estimation and video compression [1, 7]. In machine perception as well as in robotics, optical flow is used for both object detection and tracking [3, 14, 27], robot navigation and even for control of micro air vehicles [9, 17, 31].

Deployment of optical flow in real-time robotic scenarios requires low-latency processing and energy efficiency. Existing algorithms usually calculate optical flow at discrete rates based on frames obtained from conventional cameras [13]. Neuromorphic dynamic vision sensors (DVS) operate similarly to the eye's retina by providing a continuous stream of events representing brightness changes rather than absolute measurements at fixed time intervals [15]. Since optical flow computation relies on regions and time instants where brightness changes, DVS represents a viable alternative for fast optical flow prediction, demonstrated in recent works [6, 26]. Moreover, the sparsity of the events can be exploited by spiking neural networks (SNN) as opposed to artificial neural networks (ANNs). The advantage of SNNs deployed on neuromorphic hardware is low latency and energy efficiency coming from sparse computations [4].

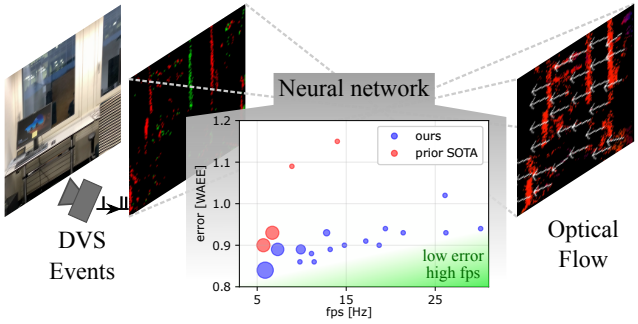

Figure 1. Optical flow estimation from DVS events: We propose a Timelens-based neural network architecture that in comparison with prior art provides lower error and higher real-time framerates.

Recently, researchers have presented an approach to produce sparse optical flow based on event data with SNNs [18]. However, there is a disconnect between large-scale architecture modelling and real-time deployments in efficient hardware. Here, we present a novel approach of a Timelens [29]-like network for sparse optical flow predictions. Apart from surpassing the optical flow baseline in terms of the average endpoint error (AEE) [18], we also address the deployment aspect through systematic model reduction and demonstrate real-time operation with a physical DVS camera, as schematically illustrated in Fig. 1.

This paper makes the following contributions:

1. We design an optical flow architecture inspired by the Timelens architecture, enriched with spiking neurons operating with DVS-based inputs.

2. We surpass the state-of-the-art self-supervised optical

flow performance for SNNs and ANNs on the MVSEC dataset [33] for event-based vision, reducing the prediction error by 6.1% in spiking, 15.6% in analog-valued spiking, and 5.5% in non-spiking mode.

3. We propose a principled methodology, involving activity and latency analysis, for reducing the network size to fit into realistic real-time hardware constraints.

4. We demonstrate model reduction from 20.4M to 0.32M parameters with 0% penalty in error with regard to the prior art [18], enabling real-time operation with DVS inputs.

2. Related Work

2.1. Deep learning of SNNs

In recent years, SNN popularity in machine learning has been increasing owing to research advancements that enabled easy modelling and training in deep learning frameworks [23, 30]. Beyond the standard Leaky Integrate-and-Fire (LIF) model, an even wider variety of neuro-inspired spiking models has been explored. In particular, a framework around so-called Spiking Neural Unit (SNU) includes the plain SNU with LIF dynamics and typical axo-dendritic synapses, as well as its variants that model further biological aspects, such as axo-axonic and axo-somatic synapses in SNUo and SNUa, respectively. These variants demonstrated improvements in large-scale speech recognition models [5]. In the context of optical flow, modifications of a LIF implementation were also proposed and called ALIF, XLIF, and PLIF [18].

2.2. Architectures for optical flow

Successful training of neural networks relies on a proper loss definition, where historically supervised losses were used [13, 16, 28]. Due to challenges with obtaining a large number of high-quality labels, it is beneficial to reformulate the training in terms of a self-supervised loss [21, 32]. In [18], the optical flow prediction task was posed as a self-supervised contrast maximization problem. This training approach can be applied to popular network architectures for optical flow prediction that include EV-FlowNet [32] and FireNet [25]. State-of-the-art SNN implementations are based on their adaptation to inputs from event-based cameras and the operation with spiking neurons [18].

2.3. Timelens

The Timelens architecture [29] was proposed in the context of event-based video frame interpolation. The design itself has been inspired from the hourglass network with skip connections for frame-based video interpolation – a problem posed initially in [19]. A peculiarity of the network architecture is the reduction of the spatial dimensionality in the encoding part using a pooling operator rather than exploiting strided convolutions. Another feature is the bigger kernel sizes for the initial two convolutions compared with the rest of the encoding/decoding blocks.

3. Network model

We propose an architecture for prediction of optical flow based on SNNs receiving an event stream from DVS. Design choices, such as spatial down- and up-sampling, channel dimensions, kernel sizes and skip connections, are inspired by the Timelens network [29]. Our network is reformulated as an SNN by incorporating spiking spatial convolutions featuring stateful neural cells and layer recurrency. An overall architectural diagram is presented in Fig. 2 and the details are described in the following subsections.

3.1. Neuron models

First, we implement an SNN using state equations that describe the common neuroscientific LIF model in a form trainable within the realm of deep learning [18, 23, 30]:

$$s_t = (1-d)(Wx_t + Hy_{t-1}) + ds_{t-1}(1-y_{t-1}) \quad (1)$$

$$y_t = h(s_t - v_{\text{th}}), \quad (2)$$

where s_t is the state – membrane potential voltage of the neuron, W and H are the input and optional recurrent weights, respectively, d is membrane potential decay factor, y_t is the output, v_{th} is a firing voltage threshold, and h is the step activation function. The model is trainable with backpropagation-through-time assuming a smooth derivative of $\text{arctanspike}(x, a) = 1/(1 + a \cdot x^2)$ for h, with $a = 10$. Trainable parameters include W, H, d and v_{th}. This neuronal model is our main focus and we will quantify architectures using it with the SNN prefix.

Secondly, we consider a more advanced biologically-inspired extension of the basic LIF – the so-called SNUo unit, which models the concept of axo-axonic synapses that enrich neuronal connectivity by modulating the neuronal outputs [5]. From implementation perspective, this leads to emission of sparse analog-valued spikes, or graded spikes in the nomenclature of Intel's Loihi 2 implementation [24]. The equations of SNUo are [5]:

$$s_t = g(Wx_t + Hy_{t-1} + ds_{t-1}(1-\tilde{y}_{t-1})) \quad (3)$$

$$\tilde{y}_t = h(s_t - v_{\text{th}}) \quad (4)$$

$$y_t = \tilde{y}_t \; o(W_o x_t + H_o y_{t-1} + b_o), \quad (5)$$

where $\tilde{y}_t$ is the unmodulated neuron output used for resetting the membrane potential, y_t is the modulated output propagating to downstream units, and g is an additional activation function that we set to leaky ReLU with a leak of 0.1. W_o, H_o and b_o are additional trainable parameters. We use

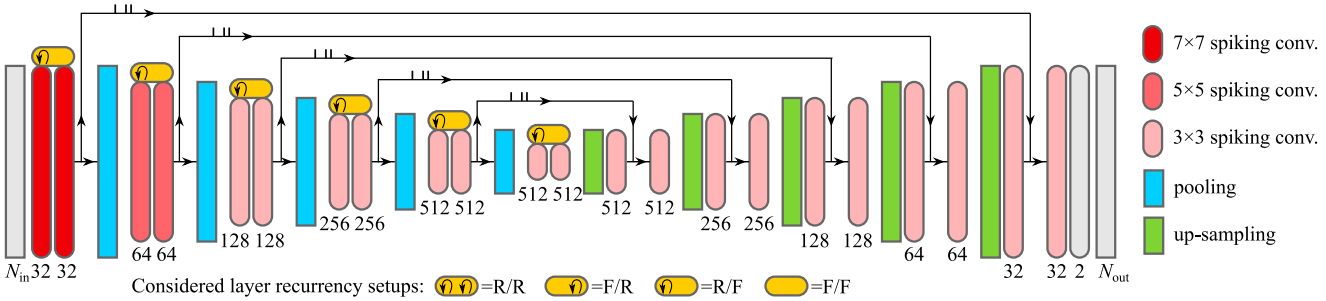

Figure 2. Spiking architecture inspired from Timelens: in the encoding part, we consider different layer-wise reccurency configurations.

the $sigmoid$ function as activation o for output modulation in Eq. 5 to mimic the inhibitory character of the axo-axonic synapses as suggested in [5]. We will quantify networks using this approach with the SNUo prefix.

Lastly, the benefits of neuromorphic internal dynamics were demonstrated also in the non-spiking mode: by operating with real-values in the so-called soft SNU (sSNU) approach [30]. The idea is to replace the step activation function h with $sigmoid$ function in Eq. 2. As sSNU operates by continuously outputting real values, we will benchmark it against non-spiking baselines. We will quantify networks using this unit with the sSNU prefix.

3.2. Network structure

Spiking convolutions work similarly to conventional convolutions found in ANNs except for the neural dynamics applied to their outputs. The computed per pixel, per channel outputs of the 2D convolutions serve as input currents (Wx_t in Eq. 1) for the spiking neural units. Simultaneously, layer-wise recurrency (Hy_{t-1} in Eq. 1) is an additional feature to capture temporal dependencies that is not always considered in SNN modelling. We explicitly mention whenever we do include this term.

The network structure is illustrated in Fig. 2. Its first stage comprises two spiking 2D convolutions expanding the N_{in} input channels to 32 output channels featuring 7×7 kernels. While the spatial dimension is retained for the spiking convolution by using stride 1 and appropriate zero padding, spatial down-sampling is performed afterwards using 2D average pooling with kernel size 2×2. The remaining encoding parts of the network are five similar encoding blocks consisting of two spiking convolutions followed by pooling operators. The kernel sizes are 3×3, except for the first encoding block (5×5). For each encoding block the number of output channels is doubled while the spatial resolution is halved. For the two spiking convolutions in each encoding block, we consider all combinations of layer-wise recurrency, as marked in Fig. 2.

For decoding, five identical decoding blocks are used. Each consists of 2D bilinear up-sampling by a factor of 2, followed by two spiking convolutions. The number of output channels gets halved with each decoding block and the convolutional kernel sizes are 3×3. Skip connections between each encoder/decoder pair of the same resolution provide values which are concatenated channel-wise before the second spiking convolution in each decoding block.

To obtain continuous optical flow values, the final layer is a 1×1 convolution with $tanh$ activation. This layer reduces 32 base channels to $N_{\text{out}} = 2$ channels representing the optical flow components u and v that correspond to horizontal and vertical optical flow magnitudes, respectively.

3.3. Input coding

The DVS event stream contains events of the form:

$$e_i = (x_i, y_i, t_i, p_i) \tag{6}$$

where x_i and y_i represent the pixel coordinates, t_i the timestamp and p_i the ON/OFF polarity of the event. Different encoding strategies have been proposed to process the raw event stream data prior to feeding it into a neural network. Commonly used input coding techniques are the count encoding [22] and the voxel grid encoding [34], depicted in Fig. 3. The count encoding loses the temporal information of single events within the aggregation window. Events get

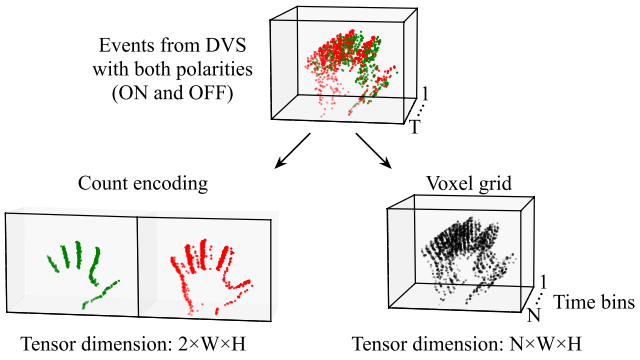

Figure 3. Different input event encodings: Count encoding (per polarity, per pixel) and voxel grid encoding via temporal bi-linear interpolation of combined events into N time bins.

accumulated per pixel and per polarity for the entire window width. On the other hand, a voxel-based representation discretizes the time span of the aggregation window and uses temporal bi-linear interpolation to populate the bins with events. Polarity is not treated as a separate channel, but negative OFF events (-1) and positive ON events (+1) are summed in a single channel.

For our spiking architectures, we opted for the voxel grid input coding. The number of discrete time bins is an additional hyperparameter. Choosing the number of bins too high yields overly sparse inputs while for a low number of bins the encoding collapses to a count representation with a single channel. In the latter case, positive and negative events can annihilate each other leading to information loss. For our spiking network, performance peaked at six time bins ($N_{\text{in}} = 6$). However, when operating in the non-spiking mode of sSNU, the count encoding with separate ON/OFF channels ($N_{\text{in}} = 2$) performed better, so we use it for sSNU-based networks. To ensure a fair comparison, the aggregation window width is fixed and the set of encoded events is therefore the same for both encoding approaches.

3.4. Training setup

All models are trained in a self-supervised fashion on the UZH-FPV Drone Racing Dataset [11], using the approach and configurations from [18]. Specifically, contrast maximization loss is applied there to compensate the motion and predict optical flow from the input events. The loss is:

$$\mathcal{L} = \mathcal{L}_{\text{contrast}}(t_{\text{ref}}^{\text{fw}}) + \mathcal{L}_{\text{contrast}}(t_{\text{ref}}^{\text{bw}}) + \lambda \mathcal{L}_{\text{smoothing}} \quad (7)$$

where contrast maximization is performed in a forward ($t_{\text{ref}}^{\text{fw}}$) as well as a backward ($t_{\text{ref}}^{\text{bw}}$) fashion w.r.t. the current reference time instance t_{ref}. $\mathcal{L}_{\text{smoothing}}$ is a Charbonnier smoothness prior [8] proposed in [32, 34] and $\lambda = 0.001$ is a balancing constant. Truncated back-propagation through time (TBPTT) is performed after every 10 forward passes.

In the original approach [18], the loss included different spatial resolutions of the optical flow maps. We analogously extended our architecture with 2D convolutions with $tanh$ activation to produce optical flow predictions of different resolutions at each decoding block. These intermediate optical flow maps are then up-sampled to the initial spatial dimension using nearest neighbour interpolation for the loss computation. Simultaneously, they are concatenated to the input channels of the subsequent decoding blocks.

However, in contrast to the prior work, we also considered an architecture with the loss applied only to the last output layer's prediction. This approach is simpler, faster and turned out to be beneficial for our architecture.

4. Simulation results

The quantitative performance and generalization abilities of the trained models (self-supervised on the UZH-FPV Drone Racing Dataset) are evaluated on the MVSEC dataset [33] following the comparison approach from [18]. The predicted sparse optical flow is compared against the ground truth optical flow provided by [32]. The ground truth labels are available at timestamps corresponding with conventional camera's frames and quantify the optical flow over one (dt = 1) or four (dt = 4) frames.

The well-established average end point error (AEE) in pixels is used to evaluate the four sequences of the dataset: outdoor_day1 (od1), indoor_flying1 (if1), indoor_flying2 (if2), indoor_flying3 (if3). For easier comparability, we introduce a weighted average endpoint error (WAEE) to combine the four metrics into a single scalar value:

$$\text{WAEE} = \left(\frac{\text{AEE}_{\text{od1}}}{w_{\text{od1}}} + \frac{\text{AEE}_{\text{if1}}}{w_{\text{if1}}} + \frac{\text{AEE}_{\text{if2}}}{w_{\text{if2}}} + \frac{\text{AEE}_{\text{if3}}}{w_{\text{if3}}}\right)/4, \quad (8)$$

where the four weights are based on the average AEE of the best-performing spiking architectures of the prior art [18] – see Supplementary Note 1 for the values for each dt setting.

Using the WAEE metric, we explored different configu-

	SNN-Timelens		sSNU-Timelens	
dt = 1	WAEE	$\overline{\%}_{\text{Outlier}}$	WAEE	$\overline{\%}_{\text{Outlier}}$
R/F	**0.84**	**4.10**	<u>0.77</u>	<u>4.23</u>
F/R	<u>0.85</u>	4.36	1.11	8.44
R/R	0.89	<u>4.26</u>	**0.73**	**3.89**
F/F	1.12	5.89	1.18	9.26
dt = 4				
R/F	**0.84**	**32.88**	<u>0.74</u>	<u>27.20</u>
F/R	<u>0.86</u>	<u>34.23</u>	1.13	44.92
R/R	0.90	35.81	**0.71**	**25.66**
F/F	1.15	52.36	1.19	48.34

Table 1. Effects of layer recurrency placement on WAEE (the lower, the better ↓) and $\overline{\%}_{\text{Outlier}}$(↓) in the encoding blocks. Best scores are in bold, while runner-ups are underlined.

Recurrency	dt = 1		dt = 4	
	WAEE	increase	WAEE	increase
R/F multi	0.92	9.52 %	0.92	9.52%
F/R multi	0.92	8.24 %	0.92	6.98%
R/R multi	0.94	5.62 %	0.95	5.56%
F/F multi	1.33	18.75 %	1.39	16.0%

Table 2. Effects of multi-layer loss function on intermediate up-sampled flow predictions for different layer recurrency placement in the encoder. WAEE(↓) and its relative increases with regard to the last layer loss in Table 1 for SNN-Timelens.

	outdoor_day1		indoor_flying1		indoor_flying2		indoor_flying3		overall	
dt = 1	AEE	$\%_{\text{Out.}}$	AEE	$\%_{\text{Out.}}$	AEE	$\%_{\text{Out.}}$	AEE	$\%_{\text{Out.}}$	WAEE	$\overline{\%}_{\text{Out.}}$
LIF-EV-FlowNet [18]	0.53	0.33	0.71	1.41	1.44	12.75	1.16	9.11	0.93	5.90
XLIF-EV-FlowNet [18]	0.45	**0.16**	0.73	0.92	1.45	12.18	1.17	8.35	0.90	5.40
LIF-FireNet [18]	0.57	0.40	0.98	2.48	1.77	16.40	1.50	12.81	1.15	8.02
PLIF-FireNet [18]	0.56	0.38	0.90	1.93	1.67	14.47	1.41	11.17	1.10	7.00
our SNN-Timelens	0.44	0.18	0.70	**0.79**	1.30	9.41	1.05	6.00	0.84	4.10
our SNUo-Timelens	**0.39**	0.17	**0.64**	0.96	**1.17**	**7.71**	**0.96**	**4.92**	**0.76**	**3.44**
EV-FlowNet [18]	0.47	0.25	0.60	**0.51**	**1.17**	**8.06**	**0.93**	5.64	0.78	**3.61**
RNN-EV-FlowNet [18]	0.56	1.09	0.62	0.97	1.20	8.82	**0.93**	**5.51**	0.83	4.10
our sSNU-Timelens	**0.36**	**0.10**	**0.58**	0.56	1.19	8.78	0.96	6.11	**0.73**	3.89
dt = 4										
LIF-EV-FlowNet [18]	2.02	18.91	2.63	29.55	4.93	51.10	3.88	41.49	0.92	35.26
XLIF-EV-FlowNet [18]	1.67	12.69	2.72	31.69	4.93	51.36	3.91	42.52	0.89	34.57
LIF-FireNet [18]	2.12	21.00	3.72	48.27	6.27	64.16	5.23	58.43	1.17	47.97
PLIF-FireNet [18]	2.11	20.64	3.44	44.02	5.94	64.02	4.98	57.53	1.11	46.55
our SNN-Timelens	1.65	11.03	2.61	29.40	4.50	50.87	3.58	40.22	0.84	32.88
our SNUo-Timelens	**1.44**	**8.98**	**2.36**	**24.18**	**3.98**	**44.71**	**3.25**	**36.01**	**0.75**	**28.47**
EV-FlowNet [18]	1.69	12.50	2.16	21.51	**3.90**	**40.72**	**3.00**	**29.60**	0.74	26.08
RNN-EV-FlowNet [18]	1.91	16.39	2.23	22.10	4.01	41.74	3.07	30.87	0.78	27.78
our sSNU-Timelens	**1.34**	**7.99**	**2.15**	**20.92**	3.97	41.31	3.17	32.44	**0.71**	**25.67**

Table 3. Evaluation on the MVSEC dataset for comparable models trained on UZH-FPV Drone Racing Dataset: AEE (the lower, the better ↓), the percentage of outliers $\%_{\text{Out.}}(\downarrow)$ per sequence, and the overall WAEE(↓) as defined in Eq. 8 as well as the average percentage of outliers $\overline{\%}_{\text{Out.}}(\downarrow)$. Best scores are in bold, while runner-ups are underlined. Horizontal lines delimit the spiking and the non-spiking models.

rations of the layer recurrency in the convolutional blocks, visualized in Fig. 2. As each block comprises two spiking convolutions, there are four different combinations of recurrent (R) and feed-forward (F) convolutions: R/F, F/R, R/R and F/F. Table 1 reports the results in terms of WAEE and the average percentage of outliers $\overline{\%}_{\text{Outlier}}$. When operating in the spiking mode, having one convolution with layer recurrency per block is favourable. In particular, best performance is achieved with recurrent layers in the first convolution (R/F). On the contrary, in the context of non-spiking mode of sSNU, double layer recurrency (R/R) is beneficial. We use these best configurations for the final models.

We also evaluated an implementation of multi-resolution loss, described in the training section. For both settings of dt $= 1$ and dt $= 4$, the reported WAEE values in Table 2 demonstrate that using the simpler setup of the loss applied only at the last layer is preferred for our architecture. A possible interpretation of the observed deterioration is that the multi-layer loss function trains the deeper decoders to encode down-sampled optical flow rather than to develop higher-level features. Furthermore, such a formulation is inconsistent with the ultimate task of the network, which is predicting high-resolution optical flow at the last layer rather than outputting the flow predictions at multiple intermediate stages. Imposing a loss only on the last layer, omits this restriction. We use this approach for all our models.

The resulting AEEs, WAEEs and outlier percentages (AEE > 3 pixels) for our Timelens-based architecture with spiking (SNN), analog-valued spiking (SNUo) and non-spiking (sSNU) units are reported in Table 3. Our model is compared with the state-of-the-art spiking and non-spiking architectures trained in the identical self-supervised setting [18]. For an extended comparison with EV-FlowNet [32,34] and Hybrid-EV-FlowNet [20] that use different training datasets and setups, see Supplementary Note 2.

For spiking neural networks, our SNN-Timelens surpasses the performance of the LIF- and XLIF-EV-FlowNet by 9.7%, 6.7% with regard to WAEE and lowers the percentage of outliers $\overline{\%}_{\text{Outlier}}$ by 30.5%, 24.1% for dt $= 1$, respectively. As the improvement over the XLIF-EV-FlowNet is 5.6% for dt $= 4$, the average prediction error is reduced by 6.1%. Table 3 shows that our SNNs are not only better on average, but outperform the comparable state-of-the-art on each MVSEC sequence for dt $= 1$ and dt $= 4$.

Operating with analog-valued spikes, SNUo-Timelens achieves a further substantial reduction in WAEE: 18.3%

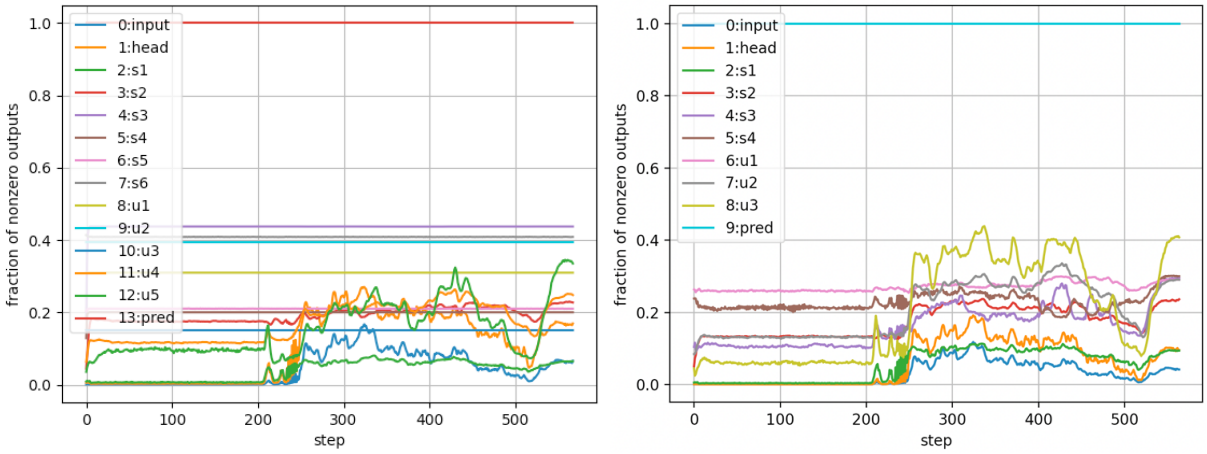

Figure 4. Spiking activity for an MVSEC test sequence: The fraction of spiking neurons is registered for all layers from input to prediction layer. Left: An architecture with 5 encoding and decoding blocks, Right: a reduced architecture with 3 encoding and decoding blocks.

and 15.6% vs. LIF- and XLIF-EV-FlowNet for dt $= 1$, respectively. Since the improvement over XLIF-EV-FlowNet for dt $= 4$ is the same, an average decrease is 15.6%. It is also remarkable that SNUo-Timelens demonstrates on-par performance with the best non-spiking self-supervised prior-art for comparable training configurations.

Lastly, the best performing model is the sSNU-Timelens incorporating neuromorphic dynamics into the non-spiking mode of operation. Despite featuring 8.5% less parameters than the best baseline EV-FlowNet (32.9M), our sSNU-TimeLens (30.1M) outperforms it with regard to WAEE for both dt $= 1$, dt $= 4$ by 6.4%, 4.1%, respectively. The average improvement over the state-of-the-art for comparable non-spiking models therefore equals 5.5%.

5. Model reduction for real-time operation

The state-of-the-art models listed in Table 3 involve tens of millions of parameters and are executed on high-end GPUs. To close the gap between large-scale architecture modelling and real-time deployment, model reduction is required. We propose a principled approach for model reduction that includes analysis of the network activity and of the relationships between the number of parameters and inference speed at different stages of the architecture.

We focus our exploration on the SNN-Timelens that could benefit the most from efficient implementation on SNN chips, such as TrueNorth [2], Loihi [10] or Kraken's SNE [12], that support the LIF equations used in the SNU. If support for analog-valued spikes increases, as in Loihi 2, the SNUo-Timelens architecture could become appealing.

5.1. Spiking activity analysis

A spiking activity analysis has been conducted to obtain potential information about the importance of different network building blocks. For a test sequence of the MVSEC dataset, the fraction of neurons that produced spikes was registered for each network layer: input layer, initial convolutional layers, encoding layers `s[i]`, decoding layers `u[i]` and the final prediction layer. Fig. 4 shows the spiking activity in SNN-Timelens architecture with 5 encoding/decoding blocks (left) compared with a network reduced to 3 encoding/decoding blocks (right). In the following we will refer to the number of encoding/decoding blocks as the number of stages of the Timelens model.

In general, the fraction of non-zero outputs, which corresponds to the fraction of neurons that spike, is almost constant until time step 210. At this time step, the drone in the DVS recording lifts off and the incoming events actually come from movement rather than static noise. The spiking activity for all layers fluctuates between 0 and 0.5 when optical flow is predicted due to the actual movement. It has to be noted that the activity of the last layer is 1.0 for all times since the final prediction layer does not feature a $step$ but rather a continuous $tanh$ activation function.

For the bigger model comprising 5 stages the fraction of non-zero outputs does not vary at all for the deep encoding `s3 - s5` and decoding `u1 - u3` layers. However, evaluation of the gradients indicates that the weights get updated during training. The question therefore arises whether these deep layers are crucial for the overall model performance. Reducing the number of stages from 5 to 3 shows indeed almost on par performance, only 2.6 % WAEE drop on MVSEC, while the spiking activity varies for all layers. The smaller model features only 1.75M parameters, which is 14.5 times less than the initial SNN architecture with 25.35M. The constant spiking activity for the layers can be interpreted as a quasi-identity mapping between early encoding and late decoding layers. Thus, dropping these layers tends to have a minor effect on the network capabilities.

5.2. Network profiling

In deep CNNs there is no simple linear relationship between the number of parameters and the inference latency. Therefore, we profiled the contributions of the components of the model to assess how the number of stages (encoding/decoding blocks) and the size of convolution impacts the inference frequency in frames per second (fps). Model performance is monitored throughout the process to find a balance between speed and quality of the predicted optical flow. The fps values are calculated from timings of 100 forward passes on 128×128 DVS inputs using Pytorch code executed on a single core of Intel Core i7 2.6GHz CPU.

Reducing channels. Network profiling has revealed that the first convolution and the first encoding block are particularly costly in terms of computations. On one hand this is due to large spatial input dimension, on the other hand it is influenced by the big convolutional kernels (7×7 and 5×5). Nevertheless, decreasing the number of output channels effectively reduces the computational costs. Fig. 5 illustrates a trade-off between the number of channels and performance in terms of WAEE and fps for the SNN-Timelens model with 5 stages. Note the non-linear relationship between convolutional channels and network parameters.

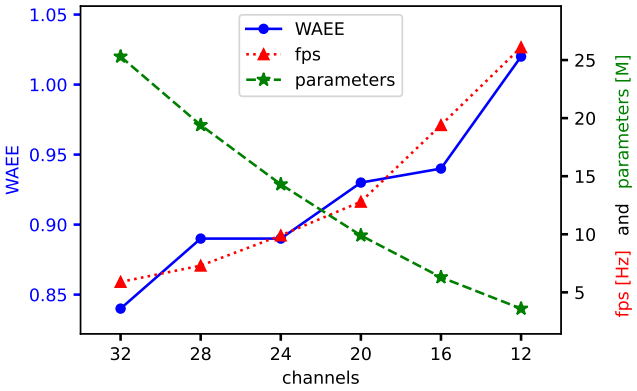

Figure 5. Impact of convolutional channels count: WAEE, network parameters (in millions [M]), and inference frequency (in frames per second [fps]) for SNN-Timelens with 5 stages.

# channels	32	28	24	20	16	12
5 stages						
WAEE	0.84	0.89	0.89	0.93	0.94	1.02
parameters	25.3	19.4	14.3	9.9	6.3	3.6
frequency	5.9	7.3	9.9	12.8	19.4	26.1
3 stages						
WAEE	0.86	0.88	0.90	0.91	0.93	1.00
parameters	1.75	1.34	0.98	0.68	0.44	0.25
frequency	9.8	11.1	14.8	17.2	26.2	32.7
2 stages						
WAEE	0.86	0.89	0.90	0.93	0.94	1.10
parameters	0.57	0.44	0.32	0.23	0.15	0.08
frequency	11.4	13.2	18.7	21.4	30.1	36.3

Table 4. Impact of convolutional channels count: WAEE, number of network parameters (in millions [M]), and inference frequency (in frames per second [fps]) of our Timelens-based SNNs for 5, 3 and 2 stages (encoding/decoding blocks).

Reducing stages. The spiking analysis showed that less than 5 stages, e.g. 3 stages, are sufficient to obtain reasonable optical flow predictions. Table 4 extends the analysis, reporting the WAEE (dt $=$ 1), the number of network parameters and model inference frequency for different number of channels and stages. Comparing the WAEE between 5 and 2 stages, we observe minor performance degradation: 0.84 versus 0.86. The 2-stage model comes with 44.4 times less parameters and increases the evaluation frequency by 93.2%. For further speedup, the number of channels of the 2-stage SNN-Timelens model can be decreased at the cost of degrading performance in terms of WAEE.

6. Model reduction results

The comparison of our architecture before and after reduction is presented for a set of selected configurations in Fig. 6. While our initial SNN-Timelens featured 5 stages with 32 channels and used 25.35M parameters, our model after reduction features only 2 stages with 32 channels and 0.57M parameters, thus reducing the number of trainable parameters by a factor of 44.4. It involves a trade-off in

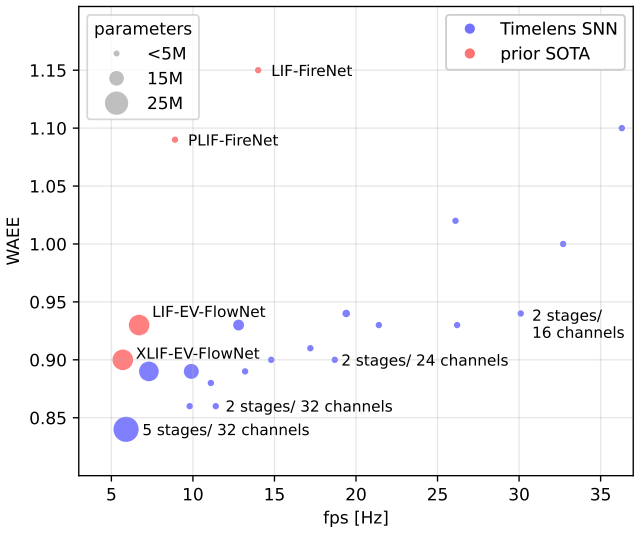

Figure 6. Model reduction results: SNN-Timelens compared with state-of-the-art (SOTA) in our CPU setup. WAEE plotted versus frames per second (fps); circle size indicates model size. For readability, only selected SNN-Timelens from Table 4 are labeled.

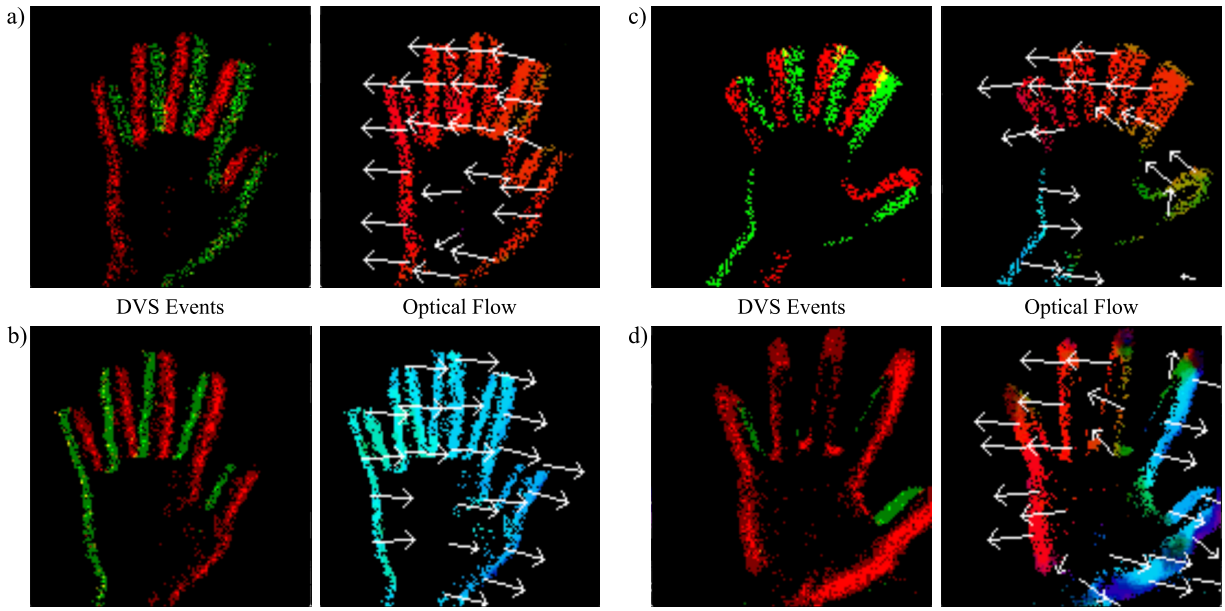

Figure 7. Real-time predictions: DVS events aggregated over one aggregation window and the corresponding optical flows from reduced SNN-Timelens (0.32M) applied for different movements of a hand: (a) to the right, (b) to the left, (c) rotation, (d) approaching the camera.

terms of WAEE performance that degrades by just 2.6% (0.84 versus 0.86). Remarkably, it still remains better than the prior state-of-the-art large models of LIF- and XLIF-EV-FlowNet (20.4M) with WAEE 0.90 and 0.93, respectively. Note that to match prior art performance (WAEE 0.90), our SNN-Timelens needs only 2 stages with 24 channels (0.32M), featuring 63.75 times less parameters.

6.1. Qualitative results

For qualitative performance assessment and validation of the generalization ability of the last proposed network with 2 stages and 24 channels, a complete real-time pipeline was implemented to process the event stream of a DVS128 camera from iniVation AG. Fig. 7 shows the optical flow predictions of different hand movements in front of the DVS. While color-coding is used to encode the optical flow, additionally a sparse arrow grid is superimposed to the optical flow for instant intuitive validation of the predictions. Arrow angles and magnitudes represent direction and magnitude of the biggest optical flow within a local 10×10 neighborhood of pixels, respectively.

The predicted flow in Fig. 7 looks reasonable and coincides with the expected dislocations caused by the moving hand. Linear motion is correctly captured (left plots) and the model generalizes well to more challenging scenarios such as rotating or approaching hand (right plots).

7. Conclusion

In this work we proposed a neuromorphic solution for optical flow estimation comprising an event camera combined with a Timelens-inspired architecture. We demonstrated SNN, SNUo and sSNU versions of our architecture, operating with different biologically inspired neuron models. By tuning the architectural design, the event encoding, the placement of recurrent connections, and the loss function formulation, we improved the performance in comparison with prior art models on the MVSEC dataset. Our architecture surpassed both SNN and ANN baselines when operating in spiking and real-valued modes, respectively. Remarkably, when operating with analog-valued spikes, it demonstrated performance comparable to the ANN baseline. Furthermore, a principled model reduction approach was proposed to meet realistic real-time hardware constraints. Our SNN-Timelens model reduced to 0.32M parameters achieves WAEE on-par with the state-of-the-art while decreasing the number of parameters by almost two orders of magnitude. Finally, a real-time pipeline was demonstrated with a physical DVS camera. Future work includes deployment of the proposed architecture on a neuromorphic SNN chip to further decrease the latency and increase energy efficiency.

8. Acknowledgements

The research was carried out in collaboration with the IBM and fortiss Center for AI (C4AI). We thank its team for discussions and assistance. The research at fortiss was supported by the HBP Neurorobotics Platform funded through the European Union's Horizon 2020 Framework Program for Research and Innovation under the Specific Grant Agreements No. 945539 (Human Brain Project SGA3).

References

[1] Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. Scale-space flow for end-to-end optimized video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. 1

[2] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, Brian Taba, Michael Beakes, Bernard Brezzo, Jente B. Kuang, Rajit Manohar, William P. Risk, Bryan Jackson, and Dharmendra S. Modha. TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(10):1537–1557, 2015. 6

[3] Sepehr Aslani and Homayoun Mahdavi-Nasab. Optical flow based moving object detection and tracking for traffic surveillance. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, 7(9):1252–1256, 2013. 1

[4] Adarsha Balaji, Anup Das, Yuefeng Wu, Khanh Huynh, Francesco G. Dell'Anna, Giacomo Indiveri, Jeffrey L. Krichmar, Nikil D. Dutt, Siebren Schaafsma, and Francky Catthoor. Mapping spiking neural networks to neuromorphic hardware. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 28(1):76–86, 2020. 1

[5] Thomas Bohnstingl, Ayush Garg, Stanislaw Wozniak, George Saon, Evangelos Eleftheriou, and Angeliki Pantazi. Speech Recognition Using Biologically-Inspired Neural Networks. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6992–6996, Singapore, Singapore, May 2022. IEEE. 2, 3

[6] Vincent Brebion, Julien Moreau, and Franck Davoine. Real-time optical flow for vehicular perception with low- and high-resolution event cameras. *IEEE Transactions on Intelligent Transportation Systems*, PP:1–13, 12 2021. 1

[7] Thomas Brox and Jitendra Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011. 1

[8] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st International Conference on Image Processing*, volume 2, pages 168,169,170,171,172, Los Alamitos, CA, USA, Nov 1994. IEEE Computer Society. 4

[9] Guido Croon, Christophe De Wagter, and Tobias Seidl. Enhancing optical-flow-based control by learning visual appearance cues for flying robots. *Nature Machine Intelligence*, 3:33–41, Jan 2021. 1

[10] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, Yuyun Liao, Chit-Kwan Lin, Andrew Lines, Ruokun Liu, Deepak Mathaikutty, Steven McCoy, Arnab Paul, Jonathan Tse, Guruguhanathan Venkataramanan, Yi-Hsin Weng, Andreas Wild, Yoonseok Yang, and Hong Wang. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro*, 38(1):82–99, Jan 2018. 6

[11] Jeffrey Delmerico, Titus Cieslewski, Henri Rebecq, Matthias Faessler, and Davide Scaramuzza. Are we ready for autonomous drone racing? the UZH-FPV Drone Racing Dataset. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6713–6719, 2019. 4

[12] Alfio Di Mauro, Moritz Scherer, Davide Rossi, and Luca Benini. Kraken: A Direct Event/Frame-Based Multi-sensor Fusion SoC for Ultra-Efficient Visual Processing in Nano-UAVs. In *2022 IEEE Hot Chips 34 Symposium (HCS)*, pages 1–19, Cupertino, CA, USA, Aug 2022. IEEE. 6

[13] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec 2015. 1, 2

[14] Bo Du, Shihan Cai, and Chen Wu. Object tracking in satellite videos based on a multiframe optical flow tracker. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(8):3043–3055, 2019. 1

[15] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jorg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-Based Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, Jan 2022. 1

[16] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-RAFT: Dense optical flow from event cameras. In *2021 International Conference on 3D Vision (3DV)*, pages 197–206. IEEE, 2021. 2

[17] Volker Grabe, Heinrich H Bülthoff, Davide Scaramuzza, and Paolo Robuffo Giordano. Nonlinear ego-motion estimation from optical flow for online control of a quadrotor UAV. *The International Journal of Robotics Research*, 34(8):1114–1135, 2015. 1

[18] Jesse Hagenaars, Federico Paredes-Vallés, and Guido de Croon. Self-supervised learning of event-based optical flow with spiking neural networks. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 2, 4, 5

[19] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[20] Chankyu Lee, Adarsh Kosta, Alex Zhu, Kenneth Chaney, Kostas Daniilidis, and Kaushik Roy. *Spike-FlowNet: Event-Based Optical Flow Estimation with Energy-Efficient Hybrid Neural Networks*, pages 366–382. Oct 2020. 5

[21] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. 2

[22] Ana I. Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso Garcia, and Davide Scaramuzza. Event-based vision

meets deep learning on steering prediction for self-driving cars. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Jun 2018. 3

[23] E. O. Neftci, H. Mostafa, and F. Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, Nov 2019. 2

[24] Garrick Orchard, E. Paxon Frady, Daniel Ben Dayan Rubin, Sophia Sanborn, Sumit Bam Shrestha, Friedrich T. Sommer, and Mike Davies. Efficient Neuromorphic Signal Processing with Loihi 2. *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 254–259, 2021. 2

[25] Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Fast image reconstruction with an event camera. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Mar 2020. 2

[26] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego. Fast event-based optical flow estimation by triplet matching. *IEEE Signal Processing Letters*, 29:2712–2716, 2022. 1

[27] A. Talukder and L. Matthies. Real-time detection of moving objects from moving vehicles using dense stereo and optical flow. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 4, pages 3718–3725, 2004. 1

[28] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 2

[29] S. Tulyakov, D. Gehrig, S. Georgoulis, J. Erbach, M. Gehrig, Y. Li, and D. Scaramuzza. Time Lens: Event-based video frame interpolation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16150–16159, Los Alamitos, CA, USA, Jun 2021. 1, 2

[30] Stanisław Woźniak, Angeliki Pantazi, Thomas Bohnstingl, and Evangelos Eleftheriou. Deep learning incorporating biologically inspired neural dynamics and in-memory computing. *Nature Machine Intelligence*, 2:325–336, Jun 2020. 2, 3

[31] Tianwei Zhang, Huayan Zhang, Yang Li, Yoshihiko Nakamura, and Lei Zhang. Flowfusion: Dynamic dense RGB-D SLAM based on optical flow. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7322–7328, 2020. 1

[32] Alex Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In *Robotics: Science and Systems XIV*. Robotics: Science and Systems Foundation, Jun 2018. 2, 4, 5

[33] Alex Zihao Zhu, Dinesh Thakur, Tolga Ozaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The Multivehicle Stereo Event Camera Dataset: An event camera dataset for 3D perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039, Jul 2018. 2, 4

[34] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth and egomotion. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1694–1694, 2019. 3, 4, 5