2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Best of Both Worlds: Hybrid SNN-ANN Architecture for Event-based Optical Flow Estimation

Shubham Negi, Deepika Sharma, Adarsh Kumar Kosta and Kaushik Roy
Elmore Family School of Electrical and Computer Engineering, Purdue University
West Lafayette, IN 47907, USA
snegi@purdue.edu

***Abstract*— In the field of robotics, event-based cameras are emerging as a promising low-power alternative to traditional frame-based cameras for capturing high-speed motion and high dynamic range scenes. This is due to their sparse and asynchronous event outputs. Spiking Neural Networks (SNNs) with their asynchronous event-driven compute, show great potential for extracting the spatio-temporal features from these event streams. In contrast, the standard Analog Neural Networks (ANNs[1]) fail to process event data effectively. However, training SNNs is difficult due to additional trainable parameters (thresholds and leaks), vanishing spikes at deeper layers, and a non-differentiable binary activation function. Furthermore, an additional data structure, "membrane potential", responsible for keeping track of temporal information, must be fetched and updated at every timestep in SNNs. To overcome these challenges, we propose a novel SNN-ANN hybrid architecture that combines the strengths of both. Specifically, we leverage the asynchronous compute capabilities of SNN layers to effectively extract the input temporal information. Concurrently, the ANN layers facilitate training and efficient hardware deployment on traditional machine learning hardware such as GPUs. We provide extensive experimental analysis for assigning each layer to be spiking or analog, leading to a network configuration optimized for performance and ease of training. We evaluate our hybrid architecture for optical flow estimation on DSEC-flow and Multi-Vehicle Stereo Event-Camera (MVSEC) datasets. On the DSEC-flow dataset, the hybrid SNN-ANN architecture achieves a 40% reduction in average endpoint error (AEE) with 22% lower energy consumption compared to Full-SNN, and 48% lower AEE compared to Full-ANN, while maintaining comparable energy usage.**

I. INTRODUCTION

Robotic systems operating at the edge require visual perception capabilities to perform various tasks, including optical flow estimation and high speed object tracking. These capabilities must be characterized by real-time processing, low energy, and robust sensing [1]. Traditional vision systems face challenges in meeting these demands, especially with high-speed motion and varying lighting conditions. Event-based cameras, such as Dynamic Vision Sensor (DVS) [2], [3], offer a promising alternative. They provide an asynchronous event stream at a high temporal resolution in the order of microseconds, high dynamic range, and extremely low power consumption [4]. However, using Analog Neural Networks (ANNs)[1] to directly process the event stream is inefficient due to the stateless computations in ANNs [5]. Event streams are handled in ANNs either by encoding timing information as input channels [6] or by specially curated input encoding techniques [7]. In contrast, machine learning architectures such as Recurrent Neural Networks (RNNs) or Long short-term memory (LSTM) do not require any input encoding and are suitable for sequential tasks [5]. But these models are more complex with larger model footprints, difficult to train, and have high computation cost [8].

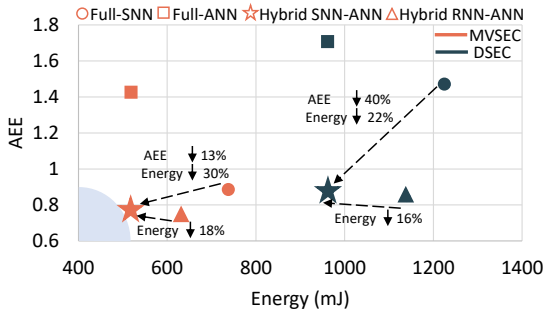

Fig. 1. AEE (lower is better) vs Inference energy for different versions of EV-FlowNet [7] on DSEC-flow and MVSEC dataset deployed on Eyeriss hardware [9]. The proposed hybrid SNN-ANN shows lower AEE and energy compared to Full-ANN, SNN and RNN[2] counterpart. The shaded blue region shows the preferred region.

Spiking Neural Networks (SNNs), which draw inspiration from biological neurons, have emerged as a promising candidate for processing sequential tasks with asynchronous and sparse event stream [10]. At the heart of an SNN is the Leaky Integrate and Fire (LIF) neuron, which can be characterized by an internal state, known as the membrane potential [11]. The membrane potential in an LIF neuron accumulates the inputs over time and decays at a certain rate governed by the leak at each timestep, emitting a spike whenever the membrane potential exceeds a specified threshold. The accumulation of inputs over time allows SNNs to learn the timing information from the input without any explicit temporal encoding, making them a special case of RNNs with less complexity.

Nonetheless, training deep SNNs remains a challenge [12] due to vanishing spikes in deeper layers, non-differentiable activations and additional threshold and leak parameters. In addition, SNN training is performed using backpropagation

[1]We refer to traditional deep learning networks that are non-recurrent in nature as ANNs due to their inputs/computations involving higher bit-precision compared to binary precision in SNNs. The underlying computing hardware is still digital.

through time (BPTT) [13] by unrolling the network in time and propagating errors, layer by layer. This unrolling over sparse activations in SNNs exacerbates vanishing gradients leading to performance degradation and slow convergence. Several solutions have been proposed over the past years to address these challenges, including learnable thresholds/leaks [5], [14], approximate surrogate gradients, [15], [16] etc. However, these solutions are still subpar compared to ANN training methodologies.

In addition to the challenges related to training, the efficiency benefits of SNNs can be best obtained when deployed (for inference) on specialized neuromorphic hardware such as IBM's TrueNorth [17], Intel's Loihi [18] etc. However, such experimental hardware architectures are yet to scale to large SNN models and are also inaccessible to most researchers. Deploying SNNs on general-purpose hardware such as GPUs, or specialized machine learning accelerators such as Eyeriss [9] turns out to be highly inefficient. This is attributed to SNNs requiring an additional data structure for the membrane potential that needs to be fetched and updated at each timestep. This results in excessive data movement and thereby, high inference energy.

To that effect, we propose a novel hybrid SNN-ANN architecture with initial SNN layers at the input, followed by several ANN layers to obtain the best of both worlds. The SNN layers with their inherent recurrence allow capturing temporal information from the sparse and asynchronous event stream. On the other hand, the ANN layers facilitate training and efficient deployment on traditional machine learning hardware. Fig. 1 depicts a comparison between Full-ANN, Full-SNN, Hybrid RNN-ANN[2] and the proposed hybrid architecture for the task of optical flow estimation. Our contributions can be summarized as follows:

- We introduce a novel hybrid SNN-ANN architecture to efficiently process the sparse spatio-temporal data from event-based cameras (Section III-C).
- We analyzed the hybrid SNN-ANN architecture (with m spiking layers), to select a network configuration optimized for both the number (m) and position of spiking layers within the network, having the best performance and training complexity for sequential tasks such as optical flow estimation (Section III-D, IV-B).
- We show that our configured hybrid architectures consistently offer lower AEE and energy for the task of optical flow estimation compared to Full-ANN, Full-SNN as well as past hybrid architectures on DSEC-flow [19] and MVSEC datasets [20] (Section IV-B, IV-C).
- We provide an estimate of the inference energy using an analytical model (Timeloop [21]) of the underlying hardware (Eyeriss [9]) that includes the cost of both data movement and computations (Section IV).

II. Related Work

In recent years, there has been an increasing interest in exploiting the high temporal resolution of event cameras

[2]First layer is ConvRNN (Fig. 4(c)) and other layers are ANN.

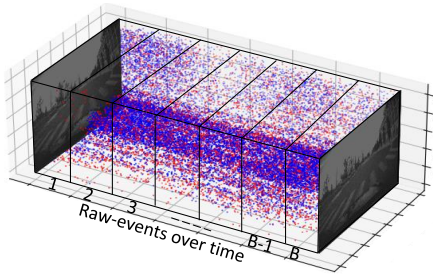

Fig. 2. Event stream between two grayscale images. The event streams are binned and fed to the networks directly.

to estimate the optical flow using neural network based approaches [22], [7], [23], [24], [25]. For instance, researchers in [7] used an ANN-based U-net [26] for optical flow estimation using self-supervised loss based on grayscale images. Recently, authors in [6] have dealt with time domain information by passing it as a channel to the first layer. With respect to lightweight architectures, authors in [27] proposed Fire-FlowNet for flow estimation. However, all these approaches require preprocessing to encode timing information in the input. Moreover, passing time domain information as input channels aggravates the computation and parameter overheads.

In regards to SNN-based architectures, authors in [24] proposed Spike-FlowNet, a hybrid version of EV-FlowNet [7], with an SNN encoder and ANN decoder. Further, authors in [25] expanded the framework in [24] by adding sensor fusion and a secondary ANN-based encoder for frame-based data. However, the reasoning behind using a full-SNN encoder was lacking in these works. To that end, in this work, we first perform an ablation study to understand the impact of the number and position of spiking layers in the hybrid architecture. We demonstrate that a well-designed hybrid SNN-ANN architecture, with the first layer as spiking and subsequent layers as ANNs, achieves superior AEE compared to previous hybrid architecture [24] on the task of optical flow estimation. Authors in [5], [14] explored trainable threshold and leak using a surrogate gradient for flow estimation. In contrast to these works, we propose a hybrid SNN-ANN architecture that utilizes the benefits of both SNNs and ANNs – SNNs help to learn temporal information, while ANNs facilitate training and enable efficient mapping to the non-spiking hardware. Further, we show that the hybrid architecture can achieve lower AEE compared to the fully spiking architecture, spiking architectures with explicit recurrence [5], and the hybrid architectures proposed in [24]. We do not compare our results with Fusion-FlowNet [25] since it uses the inputs from both the event and frame-based cameras to predict the optical flow.

III. Method

A. Sensor and Input Representation

An event camera responds to intensity changes (I) in the scene asynchronously and independently for every element in the pixel array [28]. It generates a discrete event whenever the change in log intensity exceeds a threshold

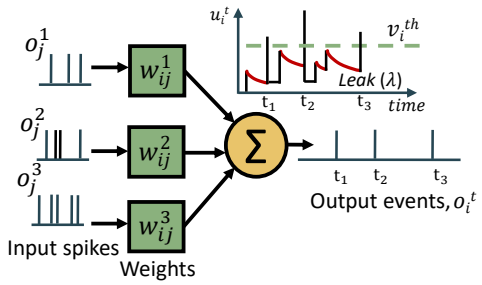

Fig. 3. Neuronal Dynamics of a LIF neuron.

(θ) i.e. $||\log(I_{t+1}) - \log(I_t)|| \geq \theta$. The discrete events are generated in the Address Event Representation (AER) format consisting of a tuple $\{x, y, t, p\}$, where x, y are the pixel coordinates; t is the timestamp; and p is the polarity of change. Stateless ANNs (i.e. non-recurrent) require encoding temporal information in the input [7], [27]. However, in this work, we directly pass inputs to all networks without encoding. For a set of N input events $\{(x_i, y_i, t_i, p_i)\}_{i\in[1,N]}$ between two consecutive grayscale images, a set of B event bins are created using bilinear sampling as shown in Fig. 2.

B. Neuron Model

Spiking neurons are fundamental units in SNNs. In this work, we use the Leaky-Integrate-and-Fire (LIF) [29] neuron which accumulates the input information over time into the neuronal state called 'membrane potential' (u). Once the membrane potential crosses a specified firing threshold(v_i^{th}), the neuron fires, generating a spike. Further, the LIF neuron also has an internal mechanism to control forgetting over time using a parameter called leak (λ). The discretized version of LIF neuron can be described as:

$$u_i^t = \lambda u_i^{t-1} + \sum_j w_{ij} o_j^t - v_i^{th} o_i^{t-1} \tag{1}$$

$$z_i^t = u_i^t / v_i^{th} - 1, o_i^t = \begin{cases} 1, & \text{if } z_i^t > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where u is the membrane potential, subscripts i and j represent the post and pre-neurons, respectively, t denotes timesteps, λ is the leak factor, w_{ij} is the weight between the i-th and j-th neurons, o is the output spike, and v_i^{th} is the threshold. The first term in Eq. (1) denotes the leakage in membrane potential, the second term computes the weighted summation of input spikes, and the third term denotes the reduction of membrane potential upon generation of an output spike at the current layer. This reduction by an amount equal to the firing threshold is termed as a 'soft reset', while a reset to zero is termed as a 'hard reset'. In this work, we use hard reset. Eq. (2) shows the spike generation mechanism in LIF neurons as depicted in Fig. 3.

C. Proposed Architecture

The proposed hybrid SNN-ANN architecture is a class of neural network architectures that incorporate both LIF and ReLU neurons in different layers, providing the advantages of both SNN and ANN architectures. The use of hybrid architecture is attributed to two main factors. Firstly, SNNs can directly process the outputs of event cameras, capturing the timing information. Secondly, ANN layers facilitate training and address the challenge of vanishing gradients in SNNs. Lastly, this architecture enables deployment on non-spiking hardware such as GPUs.

The hybrid architecture can be derived from any non-recurrent ANN for the task of optical flow estimation. We explore two types of hybrid architectures: an encoder-decoder based multi-scale architecture based on EV-FlowNet [7] and a lightweight architecture called Fire-FlowNet [27]. We evaluate several architecture sizes for the first architecture by subsequently scaling down the number of channels by a factor of 2 at each layer. Both these architectures are shown in Fig. 4. A forward pass through these architectures involves passing a 2-channeled input sequentially over T timesteps. In the encoder layer (Fig. 4 (a)) the input activation is first downsampled before passing to the next layer. Next, the output from the final encoder layer is passed through two residual blocks and four decoder layers. The decoder layers upsample the input using transposed convolution and also produce intermediate multi-scale flow predictions for each timestep. The flow prediction layers accumulate the flow over all the timesteps to generate the final full-scale flow prediction. Fire-FlowNet architecture (Fig. 4 (b)) performs similar forward passes without any downsampling/upsampling. We also compare with the corresponding RNN[2] architecture which uses the ConvRNN layer shown in Fig. 4 (c). The ConvRNN layer requires three extra convolutions, element-wise addition and tanh operation compared to the spiking layer shown in Fig. 3.

D. Optimizing Spiking Layers in Hybrid Architectures

Upon introducing the hybrid architecture, two questions need to be answered: (1) How many layers should be spiking? (2) What should be the positions of the spiking layers?

To address the first question, we explore the training complexity of hybrid architecture as we increase the number of spiking layers. The accuracy of an SNN is highly sensitive to the additional parameters associated with the spiking neuron (threshold and leak) [10], [5]. Consequently, increasing the number of spiking layers increases the training complexity making it difficult for the model to converge.

To address the second question, we consider the characteristics of the event stream and LIF neurons. Data from sensors such as DVS [3] can contain a significant amount of noise from small intensity changes in the static regions of the scene, resulting in events. This can originate from a variety of environmental factors such as reflections, edges due to shadows etc. The leak in LIF neurons serves as a low-pass filter resulting in more robust outcomes for noisy spike inputs [30]. If the deeper layers are spiking while the initial layers are non-recurrent ANNs, the noise present in inputs might not be adequately filtered out by the initial ANN layers. This could potentially lead to decreased accuracy.

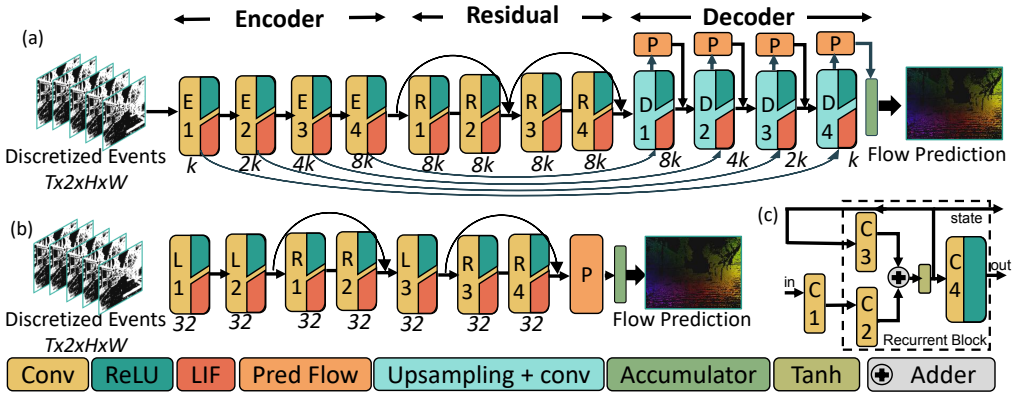

Fig. 4. Hybrid SNN-ANN architecture based on (a) EV-FlowNet [7] and (b) Fire-FlowNet [27]. The activation layers can be ReLU or LIF, depending on whether the neural network is an ANN, SNN or Hybrid SNN-ANN. Numbers below the layers show the output channels in the layer. In the input dimension, T is the timestep and 2 is the input channel because of positive and negative polarity. (c) Conv-RNN based recurrent layer [5]

Based, on this discussion we posit that a hybrid architecture with LIF neurons occupying few initial layers and all later layers being ReLU would be most effective. We will further validate these intuitions with proper experiments in section IV-B.

E. Loss Functions

To show the efficacy of hybrid architecture in learning temporal information from the input, we evaluate it on Multi-Vehicle Stereo Event Camera (MVSEC) [20] and DSEC-flow [19] datasets. We use a self-supervised loss using grayscale frames on the MVSEC dataset as it lacks reliable ground truth labels. For DSEC-flow, we utilize a supervised loss. These are described below:

Self-supervised Loss: The self-supervised loss consists of a photometric reconstruction loss ($\mathcal{L}_{photo}$) and a smoothness loss ($\mathcal{L}_{smooth}$) [31]. Photometric loss is computed using two consecutive grayscale images ($I_t(x,y), I_{t+dt}(x,y)$) and the predicted optical flow $(u_{x,y}, v_{x,y})$. It is based on the assumption that the brightness is consistent i.e. moving from pixel location (x,y) at time t to pixel location $(x+dx, y+dy)$ at time $(t+dt)$, the intensity remains the same. The predicted optical flow from the network is used to warp the second grayscale image to the first grayscale image [32] and obtain the inverse warped image $I_{t+dt}(x+u, y+v)$.

$$\mathcal{L}_{photo} = \sum_{x,y} \rho(I_t(x,y) - I_{t+dt}(x+u, y+v)) \quad (3)$$

where ρ is Charbonnier loss ($\rho(x) = (x^2+\eta^2)^r$) used for outlier rejection [33]. We set r=0.45 and η=0.001 [24].

The smoothness loss minimizes the difference in optical flow between neighboring pixels and acts as a regularizer.

$$\mathcal{L}_{smooth} = \sum_{x,y} \sum_{i,j \in \mathcal{N}(x,y)} (|u_{x,y} - u_{i,j}| + |v_{x,y} - v_{i,j}|) \quad (4)$$

where $\mathcal{N}(x,y)$ is the neighborhood of (x,y). The overall loss is:

$$\mathcal{L}_{total} = \mathcal{L}_{photo} + \lambda \mathcal{L}_{smooth} \quad (5)$$

where λ is the weight factor.

Supervised Loss: The supervised mean squared error (MSE) loss between the predicted flow (u_{pred}, v_{pred}) and the ground truth flow (u_{gt}, v_{gt}) is computed as:

$$\mathcal{L}_{supervised} = \frac{1}{K} \sum_{i=1}^{K} ((u_{pred} - u_{gt})^2 + (v_{pred} - v_{gt})^2) \quad (6)$$

where K is the number of pixels with non-zero flow.

F. Training Hybrid Architecture

The training process for SNNs and hybrid architecture differs from that of ANNs due to the non-differentiable hard threshold used as the activation function (LIF). To overcome this challenge, the training methodology proposed in [14], [15] is utilized, which involves using a surrogate gradient approximation to compute the gradients of the LIF neuron. Additionally, the network is unrolled in time for all timesteps, and the loss is backpropagated using the surrogate gradient and Backpropagation Through Time (BPTT) [13]. Apart from training the weights for SNNs and hybrid architectures, the threshold and leak parameters are also trained to enhance convergence for these networks. To get faster convergence for all the networks we also use batch normalization through time (BNTT) [34] during training.

IV. Experiments and Results

A. Experimental Setup

DSEC-Flow: The DSEC-Flow dataset [19] consists of optical flow ground truths for 24 driving sequences for a total of 7800 training samples and 2100 test samples. However, the ground truths for the test sequences are not public. Therefore we create our own test sequence by splitting the train set in an 80:20% split for the train and test set [35]. We train the networks for 200 epochs using adam optimizer [36] with a multi-step learning rate scheduler. We start with a learning rate of 0.001 and scale it by 0.7 every 10 epochs. The training is performed on a single NVIDIA A40 GPU in a supervised fashion using the loss function from Section. III-E.

MVSEC: The MVSEC dataset [20] comprises of four indoor flying sequences, two outdoor day driving sequences, and three outdoor night driving sequences. The networks are

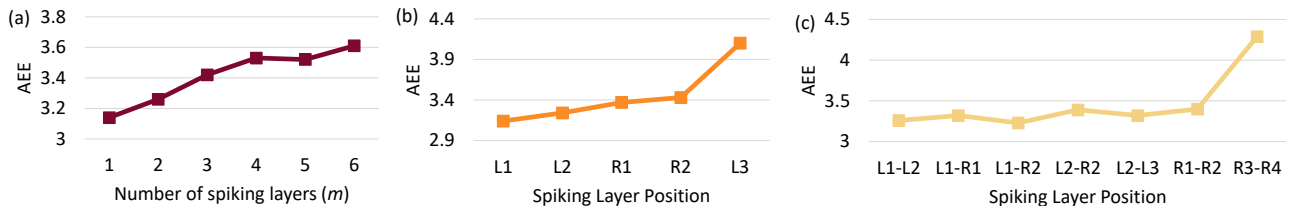

Fig. 5. Ablation study results on DSEC-flow dataset for (a) number of spiking layers (b) spiking layer position (c) spiking layer position with 2 spiking layers in the model. L1, L2, R1, R2, L3 are the layers in Fire-FlowNet architecture from Fig. 4(b).

trained in a self-supervised learning fashion. Training is performed on the outdoor_day2 driving sequence and evaluation is performed on the indoor_flying1,2,3 sequences, as well as the outdoor_day1 sequence. Transformations involving random flipping, rotation and cropping to 256x256 size are applied. The learning rate scheduler and optimizer are similar to DSEC experiments. We train for 100 epochs with an initial learning rate of 0.01. The training and evaluation are done on event volumes associated with consecutive pairs of grayscale images (termed as dt=1).

Optical Flow Evaluation Metric: To evaluate optical flow, for both datasets, we use average endpoint error (AEE). AEE measures the mean distance between the predicted and ground truth flow. This is done only for pixels containing events (n). AEE is calculated as shown below:

$$AEE = \frac{1}{n}\sum_{n} ||(u,v)_{pred} - (u,v)_{gt}||_2 \tag{7}$$

Energy Estimation: Several recent works for optical flow estimation [24], [25], [14] include preliminary evaluation of ANN and SNN inference energy based on [37]. This evaluation provides an incomplete analysis as it compares the computation energy but does not account for the cost of data movement, a significant contributor to inference energy in modern machine learning workloads [38]. Therefore, for our analysis, we use an analytical performance architecture, Timeloop [21], which incorporates both communication and computation energies. We adopt the same approach as in the study by [39] and map our networks to a specialized ML accelerator, Eyeriss [9] to obtain the energy estimates. For spiking layers, as part of computation energy, only accumulator energy is considered as opposed to multiply-and-accumulate energy for ANNs. We used 45nm technology node for all our energy calculations.

B. DSEC-flow Results

Ablation Study: We perform an ablation study to verify our hypothesis (Section III-D) on the number and position of spiking layers in the hybrid architecture. We start with FireFlowNet architecture from Fig. 4(b) and evaluate it on the DSEC-flow dataset. As depicted in Fig. 5(a), we observe that the AEE for the FireFlowNet architecture increases as the number of spiking layers increases. We also conducted experiments with two spiking layers, exploring various layer positions (Fig. 5 (c)). The AEE for all the cases with two spiking layers is higher compared to the AEE with just one spiking layer. Moreover, in both single and two spiking layer cases, shifting these layers deeper into the network resulted in an increased AEE (Fig. 5 (b) and (c)).

All these ablations are following our hypothesis from Section III-D. Based on these findings, we conclude that keeping the first layer as a spiking layer in the hybrid architecture gives the lowest AEE due to reduced training complexity. For all the forthcoming experiments, the first layer in the hybrid architecture is assigned to be the spiking layer, and the remaining layers consist of neurons with ReLU activation.

TABLE I

AEE (LOWER IS BETTER) ON DSEC-FLOW. BEST IN BOLD.

Network Architecture	AEE	Energy (mJ)
[35] (LSTM)	1.28	-
EV-FlowNet [40]	2.32	-
E-RAFT [40]	**0.79**	-
Base-Full-ANN	1.71	962.13
Base-Full-SNN	1.47	1228.60
Base-Hybrid (ours)	0.88	961.39
Mini-Full-ANN	1.77	338.93
Mini-Full-SNN	1.65	508.96
Mini-Hybrid (ours)	0.97	338.34
Micro-Full-ANN	1.93	80.39
Micro-Full-SNN	1.76	144.26
Micro-Hybrid (ours)	1.10	**79.96**
FireFlowNet-Full-ANN	6.56	352.74
FireFlowNet-Full-SNN	3.76	460
FireFlowNet-Hybrid (ours)	3.14	356.21

Experiments with Network Size: In this study, we explore the effect of reducing the number of base channels (k) in the architecture from Fig. 4(a), and assess the performance of different network sizes using the training methodology from section IV-A. We experiment with several values for k, such as Base (k=64), Mini (k=32) and Micro (k=16). Additionally, we compare our hybrid architecture with a Full-SNN (m=total number of layers), and Full-ANN (m=0), where m is the number of spiking layers.

Full-SNN vs Full-ANN: The results presented in Table I show that all the Full-SNN architectures (Base - FireFlowNet) give lower AEE compared to the Full-ANN version. Specifically, Base-Full-SNN has 14% lower AEE compared to Base-Full-ANN. This improvement can be attributed to the membrane potentials in SNNs that enable them to learn the temporal information from the input. In contrast, ANNs being stateless, struggle to capture this temporal aspect since it is not encoded directly in the input. However, SNNs consume more inference energy compared to ANNs due to the overhead of fetching and updating the membrane potentials.

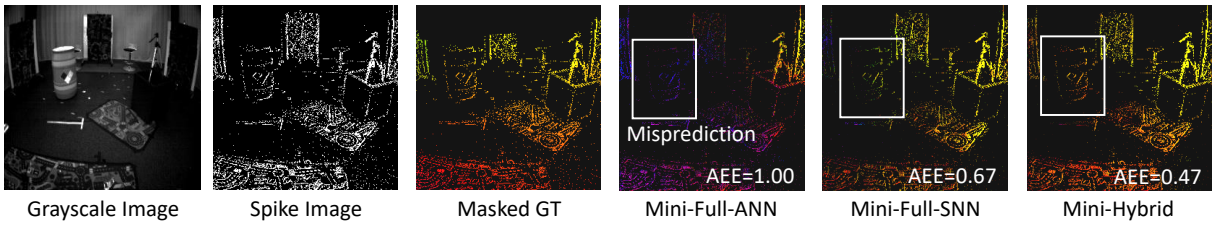

Fig. 6. Qualitative results of best performing Full-ANN, SNN and Hybrid SNN-ANN on indoor_flying1 sequence from MVSEC dataset.

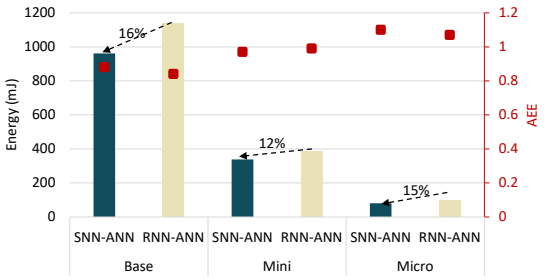

Fig. 7. AEE and Energy comparison to hybrid RNN-ANN[2].

Hybrid vs Full-SNN, Full-ANN: Furthermore, we observe that the hybrid architecture outperforms the Full-ANN and Full-SNN architectures in terms of AEE. In addition to the low AEE, it also gives comparable inference energy to the Full-ANN architecture. Specifically, Base-Hybrid has 48% lower AEE compared to Base-Full-ANN and 1.3× lower energy than Base-Full-SNN. The lower AEE can be attributed to the fact that the spiking layer in the hybrid architecture can learn the temporal information from the input and also facilitate the optimizer to converge faster due to fewer trainable parameters. Additionally, the hybrid architecture consumes less energy than the Full-SNN architecture, because it has fewer spiking layers and therefore less overhead from membrane potentials. Another interesting observation is comparing architectures of different sizes. For instance, Micro-Hybrid has 35% lower AEE and 12× lower energy compared to Base-Full-ANN. Compared to Base-Full-SNN, Micro-Hybrid has 25% lower AEE and 15× lower energy.

Hybrid vs state-of-the-art (SOTA): When compared to SOTA architectures for the DSEC flow dataset, our proposed Base-Hybrid architecture outperforms most of them. For example, in [35], authors use LSTM layers instead of convolutional layers in the EV-FlowNet architecture, but our Base-Hybrid architecture shows a 31% lower AEE than [35]. However, E-RAFT gives 12% lower AEE compared to Base-Hybrid which can be attributed to the Gated Recurrent Unit and iterative flow refinement in E-RAFT. Note, however, that the complex architecture of E-RAFT would lead to higher inference energy compared to Hybrid architecture.

Hybrid SNN-ANN vs Hybrid RNN-ANN: We also compare the hybrid SNN-ANN architecture with a corresponding hybrid RNN-ANN architecture which uses the ConvRNN layer (Fig. 4(c)) as the first layer and all the other layers are ANN-like in EV-FlowNet architecture. We can see from Fig. 7 that as we reduce the model size, the AEE for the hybrid architecture is close to RNN with nearly 15% lower energy. This can be attributed to the simpler inherent recurrence provided by SNNs compared to RNNs.

C. MVSEC Results

We conducted experiments on the MVSEC dataset, using the same architectures employed for the DSEC-flow dataset. The quantitative results are presented in Table II and visualized in Fig. 6.

Hybrid vs Full-SNN, Full-ANN: The hybrid architecture outperforms the Full-ANN and Full-SNN architectures in terms of AEE and inference energy. Specifically, the Mini-Hybrid showed 47% and 25% lower average AEE compared to the Full-ANN and Full-SNN versions, respectively. Furthermore, Mini-Hybrid demonstrated a 32% reduction in inference energy compared to Mini-Full-SNN and had comparable energy consumption to Mini-Full-ANN. The ease of training the hybrid architecture contributes to the AEE improvement. Additionally, the reduction in inference energy compared to Full-SNN is due to the fewer spiking layers in the hybrid architectures

Hybrid vs SOTA: Mini-Hybrid, Mini-Full-SNN has a lower average AEE compared to the baselines that encode temporal information in the input [27], [7]. We attribute these improvements to the LIF neurons, which help to learn temporal information better compared to ANN architectures. In comparison to Spike-FlowNet [24], which uses all the encoder layers as spiking layers, Mini-Hybrid has a 17% lower average AEE. This finding is consistent with our ablation study, where we observed that the AEE tends to increase with an increase in the number of spiking layers (Fig. 5 (a)). Further, compared to Fusion-FlowNet [25] Mini-Hybrid has only 6% higher AEE. However, Fusion-FlowNet uses inputs from both the frame and event camera. In addition, we found that Base-Full-SNN and Mini-Hybrid achieve 7% and 23% lower average AEE, respectively than the fully spiking architecture XLIF-EV-FlowNet. We also compared the performance of the lightweight fully spiking architecture XLIF-FireNet [5] with our FireFlowNet-Full-SNN and Hybrid architecture and found that they achieve 6% and 12% lower average AEE, respectively. This performance of FireFlowNet-Full-SNN and FireFlowNet-Hybrid can be attributed to the presence of the BNTT [34] and low training complexity in these architectures.

TABLE II

AEE AND ENERGY RESULTS ON MVSEC DATASET. WE REPORT AVERAGE AEE SINCE THERE ARE MULTIPLE TEST SETS. BEST IN BOLD. FUSION-FLOWNET USES INPUT FROM BOTH THE EVENT AND FRAME CAMERA.

Network Architecture	outdoor_day1	indoor1	indoor2	indoor3	Average AEE	Energy (mJ)
EV-FlowNet [7]	0.49	1.03	1.72	1.53	1.19	-
Back to Event Basics$_{\text{EVF}}$ [27]	0.92	0.79	1.40	1.18	1.07	-
Back to Event Basics$_{\text{FIRE}}$ [27]	1.06	0.97	1.67	1.43	1.28	-
XLIF-EV-FlowNet [5]	0.45	0.73	1.45	1.17	0.95	-
XLIF-FireNet [5]	0.54	0.98	1.82	1.54	1.22	-
Spike-FlowNet [24]	0.49	0.84	1.28	1.11	0.93	-
Fusion-FlowNet [25]	0.59	**0.56**	**0.95**	**0.76**	**0.72**	-
Base-Full-ANN	0.51	1.24	2.11	1.87	1.43	518.58
Base-Full-SNN	0.51	0.77	1.21	1.05	0.88	738.38
Base-Hybrid (ours)	**0.4**	0.69	1.05	0.92	0.77	518.21
Mini-Full-ANN	0.56	1.23	2.13	1.90	1.45	207.63
Mini-Full-SNN	0.53	0.89	1.40	1.25	1.02	303.12
Mini-Hybrid (ours)	0.41	0.69	1.04	0.92	0.77	207.27
Micro-Full-ANN	0.74	1.24	2.14	1.90	1.50	49.04
Micro-Full-SNN	0.61	0.89	1.36	1.20	1.02	87.06
Micro-Hybrid (ours)	0.42	0.79	1.20	1.07	0.87	**48.78**
FireFlowNet-Full-ANN	1.15	1.35	2.20	1.96	1.66	194.39
FireFlowNet-Full-SNN	0.67	0.99	1.58	1.36	1.15	211.91
FireFlowNet-Hybrid (ours)	0.59	0.91	1.50	1.27	1.07	196.29

V. DISCUSSION

Previous studies often reported lower accuracy for SNNs compared to ANNs [41], [42], [43]. These studies typically presented results on static datasets like CIFAR-10, CIFAR-100 [44] and ImageNet [45]. Optical flow estimation, which requires capturing temporal information from the input, benefits from SNNs' inherent recurrence (membrane potential). This enables SNNs to surpass ANNs in learning such temporal information while utilizing fewer parameters than RNNs. Furthermore, it is notable that inference energy is dominated by both computation and data movement [38]. Nonetheless, prior studies solely compared the computational energy of SNNs and ANNs. Therefore, for a fair comparison in this work, we consider the energy stemming from both computation and data movement. We observe that SNNs have nearly 30-40% higher inference energy compared to ANNs when deployed on non-spiking hardware (Eyeriss). On the other hand, RNNs incur even more energy costs due to their considerably more complex computations. In light of these observations, hybrid SNN-ANN architecture offers an appealing middle ground (both for GPU implementation and training cost) by presenting itself as a low-energy alternative to Full-SNNs and RNNs while retaining their capabilities of processing temporal inputs effectively.

VI. CONCLUSION

In conclusion, this work presents a new approach to learning temporal information from event-inputs by introducing hybrid SNN-ANN architecture. Hybrid architectures benefit from the strengths of both SNN and ANN layers and can effectively learn the temporal information from inputs, without requiring any explicit input encoding. Our study demonstrates that a well-designed hybrid SNN-ANN architecture, with a spiking layer as the first layer and subsequent layers as ANNs, achieves better AEE compared to past hybrid architectures with multiple spiking layers. Additionally, this hybrid architecture alleviates the challenges of its deployment on non-spiking hardware (GPUs and custom ML hardware). Extensive quantitative and qualitative evaluations conducted on two commonly used optical flow datasets (DSEC-flow and MVSEC) demonstrate the superiority of hybrid architecture compared to state-of-the-art models. Finally, our deployment on non-spiking ML hardware shows that the inference energy of hybrid architecture is better than SNNs and RNNs and is comparable to ANNs. We believe our work presents a promising direction for the development of efficient and accurate architectures with low training complexity and seamless deployment on traditional machine learning hardware.

ACKNOWLEDGMENTS

This work was partly supported by the IARPA MicroE4AI program, the Center for the Co-Design of Cognitive Systems (COCOSYS), a DARPA-sponsored JUMP center of Semiconductor Research Corporation (SRC).

REFERENCES

[1] A. Baxi, M. Eisen, S. Sudhakaran, F. Oboril, G. S. Murthy, V. S. Mageshkumar, M. Paulitsch, and M. Huang, "Towards factory-scale edge robotic systems: Challenges and research directions," *IEEE Internet of Things Magazine*, vol. 5, no. 3, pp. 26–31, 2022.

[2] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb 2008.

[3] C. Brandli *et al.*, "A 240 × 180 130 db 3 µs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, Oct 2014.

[4] C. Posch *et al.*, "Retinomorphic event-based vision sensors: bioinspired cameras with spiking output," *Proceedings of the IEEE*, vol. 102, no. 10, pp. 1470–1484, 2014.

[5] J. Hagenaars *et al.*, "Self-supervised learning of event-based optical flow with spiking neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7167–7179, 2021.

[6] A. Z. Zhu *et al.*, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 989–997.

[7] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Ev-flownet: Self-supervised optical flow estimation for event-based cameras," *arXiv preprint arXiv:1802.06898*, 2018.

[8] W. Ponghiran *et al.*, "Hybrid analog-spiking long short-term memory for energy efficient computing on edge devices," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2021, pp. 581–586.

[9] Y.-H. Chen *et al.*, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE journal of solid-state circuits*, vol. 52, no. 1, pp. 127–138, 2016.

[10] N. Rathi, I. Chakraborty, A. Kosta, A. Sengupta, A. Ankit, P. Panda, and K. Roy, "Exploring neuromorphic computing based on spiking neural networks: Algorithms to hardware," *ACM Computing Surveys*, 2022.

[11] K. Roy *et al.*, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, 2019.

[12] N. Rathi *et al.*, "Exploring spike-based learning for neuromorphic computing: Prospects and perspectives," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2021, pp. 902–907.

[13] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[14] A. K. Kosta and K. Roy, "Adaptive-spikenet: event-based optical flow estimation using spiking neural networks with learnable neuronal dynamics," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 6021–6027.

[15] C. Lee *et al.*, "Enabling spike-based backpropagation for training deep neural network architectures," *Frontiers in neuroscience*, p. 119, 2020.

[16] N. Rathi and K. Roy, "Diet-snn: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[17] M. V. DeBole *et al.*, "Truenorth: Accelerating from zero to 64 million neurons in 10 years," *Computer*, vol. 52, no. 5, pp. 20–29, 2019.

[18] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *Ieee Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[19] M. Gehrig *et al.*, "Dsec: A stereo event camera dataset for driving scenarios," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4947–4954, 2021.

[20] A. Z. Zhu *et al.*, "The multivehicle stereo event camera dataset: An event camera dataset for 3d perception," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2032–2039, 2018.

[21] A. Parashar *et al.*, "Timeloop: A systematic approach to dnn accelerator evaluation," in *2019 IEEE international symposium on performance analysis of systems and software (ISPASS)*. IEEE, 2019, pp. 304–315.

[22] S. Meister *et al.*, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[23] C. Ye *et al.*, "Unsupervised learning of dense optical flow, depth and egomotion with event-based sensors," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5831–5838.

[24] C. Lee, A. K. Kosta, A. Z. Zhu, K. Chaney, K. Daniilidis, and K. Roy, "Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks," in *European Conference on Computer Vision*. Springer, 2020, pp. 366–382.

[25] C. Lee *et al.*, "Fusion-flownet: Energy-efficient optical flow estimation using sensor fusion and deep fused spiking-analog network architectures," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6504–6510.

[26] O. Ronneberger *et al.*, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[27] F. Paredes-Vallés *et al.*, "Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3446–3455.

[28] G. Gallego *et al.*, "Event-based vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 154–180, 2020.

[29] L. F. Abbott, "Lapicque's introduction of the integrate-and-fire model neuron (1907)," *Brain research bulletin*, vol. 50, no. 5-6, pp. 303–304, 1999.

[30] S. S. Chowdhury *et al.*, "Towards understanding the effect of leak in spiking neural networks," *Neurocomputing*, vol. 464, pp. 83–94, 2021.

[31] J. J. Yu *et al.*, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*. Springer, 2016, pp. 3–10.

[32] M. Jaderberg *et al.*, "Spatial transformer networks," *Advances in neural information processing systems*, vol. 28, 2015.

[33] D. Sun *et al.*, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *International Journal of Computer Vision*, vol. 106, pp. 115–137, 2014.

[34] Y. Kim *et al.*, "Revisiting batch normalization for training low-latency deep spiking neural networks from scratch," *Frontiers in neuroscience*, p. 1638, 2021.

[35] W. Ponghiran *et al.*, "Event-based temporally dense optical flow estimation with sequential neural networks," *arXiv preprint arXiv:2210.01244*, 2022.

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

[37] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE, 2014, pp. 10–14.

[38] H. Kwon *et al.*, "Understanding reuse, performance, and hardware cost of dnn dataflow: A data-centric approach," in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2019, pp. 754–768.

[39] D. Sharma *et al.*, "Identifying efficient dataflows for spiking neural networks," in *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, 2022, pp. 1–6.

[40] M. Gehrig *et al.*, "E-raft: Dense optical flow from event cameras," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 197–206.

[41] N. Rathi, G. Srinivasan, P. Panda, and K. Roy, "Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation," *arXiv preprint arXiv:2005.01807*, 2020.

[42] S. S. Chowdhury *et al.*, "Towards ultra low latency spiking neural networks for vision and sequential tasks using temporal pruning," in *European Conference on Computer Vision*. Springer, 2022, pp. 709–726.

[43] Y. Kim, Y. Li, H. Park, Y. Venkatesha, A. Hambitzer, and P. Panda, "Exploring temporal information dynamics in spiking neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 7, 2023, pp. 8308–8316.

[44] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images," 2009.

[45] J. Deng *et al.*, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.