SNN-Driven Event-Based Flow and Rotation Estimation with SO(3) Refinement

Ruimin Sun[1], **Haoran Xu**[1], **De Ma**[1,2*]

[1]College of Computer Science and Technology, Zhejiang University, China
[2]The State Key Lab of Brain-Machine Intelligence, Zhejiang University, China
{sunruimin, xu_hn, made}@zju.edu.cn

Abstract

Spiking Neural Networks (SNNs) offer a promising direction for energy-efficient event-based vision by leveraging sparse, temporally precise spikes. We propose a directly trained, fully spiking model for optical flow estimation, featuring a novel Spike GRU and membrane potential carryover for improved temporal modeling. On the DSEC-Flow benchmark, our model achieves competitive accuracy while reducing energy consumption by 42.88× over EV-FlowNet and 38× over TIDNet. Building on the predicted motion field, we infer camera rotation and, to the best of our knowledge, are the first to construct panoramic event images from SNN-based flow. We further introduce an optional unsupervised $SO(3)$ refinement step that improves rotation accuracy by maximizing panorama consistency—without IMU or pose supervision. Our results achieve comparable visual quality to CMax-SLAM, showing that SNNs can enable fast and high-level spatial perception using only event-based input.

Introduction

Event cameras have emerged as a promising alternative to traditional frame-based sensors, offering advantages such as microsecond-level temporal resolution ($1\ \mu s\ =\ 10^{-6}\ s$), high dynamic range (60–140 dB), and ultra-low power consumption (as low as $10\ mW$). Unlike conventional cameras that capture full image frames at fixed intervals, event cameras asynchronously record per-pixel brightness changes. These properties enable effective operation under high-speed motion or extreme lighting (Gallego et al. 2020; Bouwmeester, Paredes-Vallés, and De Croon 2023).

To leverage the rich spatiotemporal signals of event streams, recent works have applied artificial neural networks (ANNs) to tasks such as optical flow and depth estimation (Zhu et al. 2018a; Gehrig et al. 2021). While effective, ANNs are originally designed for dense, frame-based images, often relying on compute- and memory-intensive operations. This makes them less compatible with the binary, asynchronous and sparse nature of event data.

Spiking neural networks (SNNs) offer a brain-inspired alternative, transmitting and computing with sparse binary spikes. Their event-driven, temporally-aware operations align naturally with the characteristics of event data (Kosta and Roy 2023; Tian and Andrade-Cetto 2024). When deployed on neuromorphic hardware such as Loihi (Davies et al. 2018), TrueNorth (Akopyan et al. 2015), or Darwin (Ma et al. 2024), SNNs can achieve orders-of-magnitude lower energy and latency, making them attractive for edge or real-time applications.

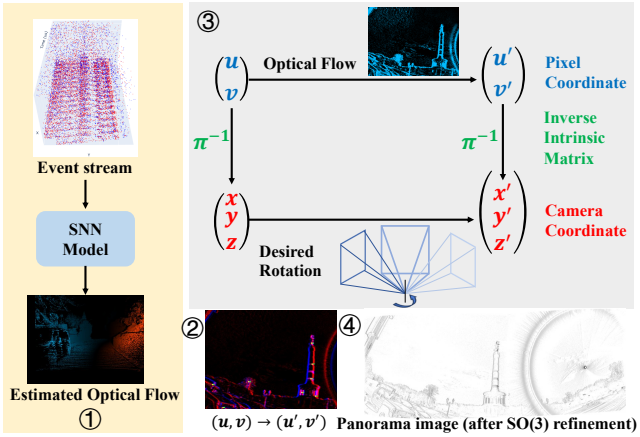


Figure 1: Overview of our pipeline. Event stream → ① SNN predicts optical flow → ② Optical flow provides point correspondences before and after rotation → ③ Camera rotation estimation → ④ Panorama image construction. The full $SO(3)$ refinement process is detailed in Fig.5. A video illustration of the entire pipeline is included in the Appendix.

Despite these advantages, existing SNN models for optical flow estimation still lag behind ANN counterparts in accuracy. A key challenge is the lack of mechanisms to capture long-term motion continuity. To address this, we propose a spike-based gated recurrent unit (Spike GRU) that introduces biologically plausible recurrence into the SNN framework. In addition, we extend the temporal memory of Leaky Integrate-and-Fire (LIF) neurons by carrying over membrane potentials across adjacent segments, allowing the model to learn over longer temporal windows.

We build upon the lightweight ANN model TIDNet (Wu, Paredes-Vallés, and De Croon 2024) and fully convert it into a spiking architecture for energy-efficient optical flow es-

*Corresponding author.

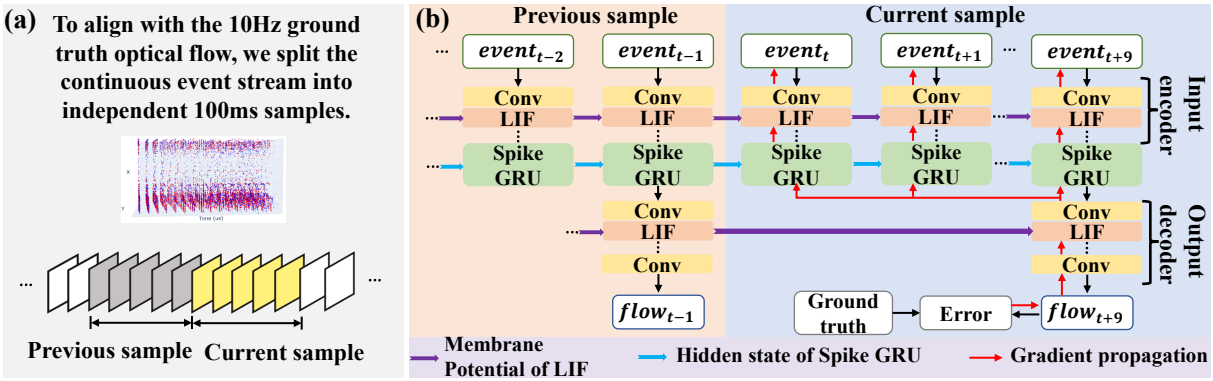


Figure 2: Overview of SNN-based optical flow estimation. (a) The event stream is divided into 100ms segments to match 10Hz supervision. (b) LIF and Spike-GRU modules process events sequentially, with membrane potentials and hidden states carried over across segments to preserve temporal continuity.

timation. Beyond flow, we are the **first to leverage SNN-predicted optical flow to construct panoramic event images**, by inferring camera rotation from the predicted motion field. This enables temporally consistent, compact representations of the scene from sparse motion-compensated events. Such global views benefit higher-level tasks such as SLAM (Pellerito et al. 2024) or loop closure (Lipson, Teed, and Deng 2024).

Importantly, the rotation matrices predicted by our SNN model require no IMU data or external supervision, and the inference runtime is extremely low, making it well-suited for real-time deployment. While we also introduce an optional $SO(3)$ refinement step to further improve accuracy (where $SO(3)$ refers to the group of all 3D rotation matrices), the core SNN pipeline alone already delivers strong rotation estimation performance with minimal latency.

Our main contributions are summarized as follows:

- We propose a directly trained, energy-efficient SNN model for event-based optical flow estimation, enhanced by a novel Spike GRU and inter-segment membrane potential propagation.
- To the best of our knowledge, this is the first SNN-based pipeline that estimates camera rotation from event flow to construct panoramic representations without additional sensors.
- We present an unsupervised $SO(3)$ refinement strategy that improves rotation estimation by maximizing panorama consistency, entirely independent of IMU data.

Preliminary

ConvGRU (Ballas et al. 2015) extends GRU to spatiotemporal data by replacing matrix multiplications with convolutions. It employs three gates: an update gate $\mathcal{Z}_t$ to control memory retention, a reset gate $\mathcal{R}_t$ for modulating prior state influence, and an output gate that computes the current hidden state $\mathcal{H}_t$ (σ denotes the sigmoid activation function):

$$\mathcal{Z}_t = \sigma(\mathrm{Conv}([\mathcal{H}_{t-1}, \mathcal{X}_t])), \tag{1}$$

$$\mathcal{R}_t = \sigma(\mathrm{Conv}([\mathcal{H}_{t-1}, \mathcal{X}_t])), \tag{2}$$

$$\tilde{\mathcal{H}}_t = \tanh(\mathrm{Conv}([\mathcal{R}_t * \mathcal{H}_{t-1}, \mathcal{X}_t])), \tag{3}$$

$$\mathcal{H}_t = (1 - \mathcal{Z}_t) * \mathcal{H}_{t-1} + \mathcal{Z}_t * \tilde{\mathcal{H}}_t. \tag{4}$$

While widely used in ANNs, ConvGRU has not been explored in spiking networks. In this work, we propose a spike-based variant to enable temporal recurrence in SNNs.

Leaky Integrate-and-Fire (LIF) Neuron Model. The LIF model (Gerstner et al. 2014) is a standard spiking neuron model where the membrane potential μ_l^t evolves over time by integrating inputs and leaking past activity:

$$\mu_l^t = \lambda_l \mu_l^{t-1} + \mathrm{Conv}(\mathcal{O}_{l-1}^t) - \mathcal{V}_l \mathcal{O}_l^{t-1}. \tag{5}$$

A spike $\mathcal{O}_l^t$ is generated if the normalized potential exceeds the threshold:

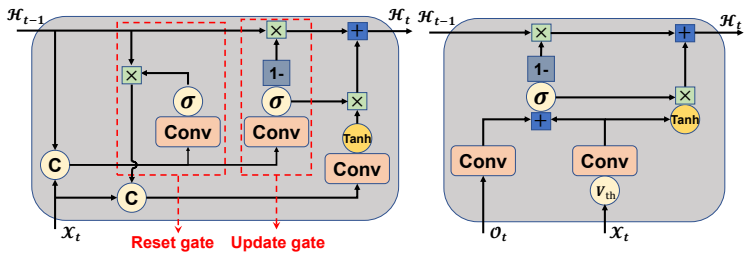
$$\mathcal{Z}_l^t = \frac{\mu_l^t}{\mathcal{V}_l} - 1, \quad \mathcal{O}_l^t = \begin{cases} 1, & \mathcal{Z}_l^t > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

We follow Adaptive-SpikeNet (Kosta and Roy 2023) by treating the threshold $\mathcal{V}_l$ and decay rate λ_l as channel-wise trainable parameters, allowing the model to adaptively learn temporal dynamics from event data.

Proposed Methods

SNN-Based Optical Flow Estimation

To align with the 10Hz ground-truth optical flow supervision, we divide the event stream into non-overlapping 100ms segments, as illustrated in Fig.2(a). For each segment, a spiking neural network (SNN) predicts the corresponding optical flow. Unlike typical SNN applications in image classification or segmentation, which operate on short and independent samples, our task deals with temporally dense, continuous event streams that span up to tens of minutes per sequence. This long-duration setting requires architectures that retain temporal context beyond the conventional sample-by-sample processing.



	FLOPs (single Conv FLOPs = $c_{in} \times c_{out} \times H \times W \times k^2 = 160 \times 96 \times 48 \times 64 \times 9 = 0.42$G)	Energy (ANN: FLOPs $\times E_{MAC}$ SNN: FLOPs $\times E_{AC} \times T \times F_r$)
Conv GRU	$3 \times 0.42 = 1.26$ G	$3 \times 0.42 \times 4.6 = 5.796$ pJ
Spike GRU	$2 \times 0.42 = 0.84$ G	$2 \times 0.42 \times 0.9 \times 1 \times 0.17 = 0.129$ pJ

Figure 3: Comparison of architectures and energy efficiency. (Left) Standard ConvGRU with reset and update gates. (Middle) Proposed Spike GRU, which uses event-dCameraReadyriven computations and binary spikes for lightweight gating. (Right) FLOPs and energy consumption comparison; see experiments for full details.

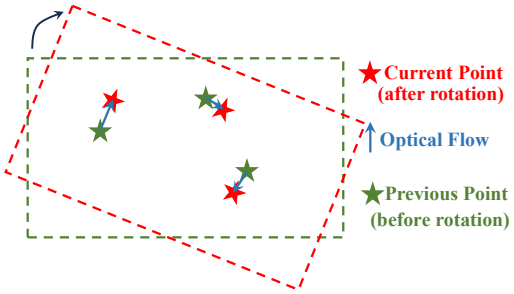


Figure 4: Illustration of selected event points (green stars) and their displaced positions (red stars) using the SNN-predicted optical flow. Blue arrows indicate motion before and after camera rotation.

We adopt the TIDNet (Wu, Paredes-Vallés, and De Croon 2024) backbone and modify it with spiking components. As shown in Fig. 2(b), we insert Leaky Integrate-and-Fire (LIF) neurons at both the input encoder and output decoder stages to capture membrane potential dynamics. Unlike in previous SNN works, where the membrane potential is reset between input samples, we carry the potential μ_l^t from one 10ms segment to the next. This strategy enables better temporal continuity and long-term information preservation. It also enhances prediction consistency across time by leveraging the recurrent nature of SNNs. Furthermore, as LIF neurons only emit spikes when their membrane potential crosses a firing threshold, the model is more energy efficient. It is important to note that gradients are only backpropagated within each current 100ms segment, without affecting the membrane states of previous segments.

To replace the floating-point-heavy ConvGRU commonly used in ANN-based designs, we propose a spike-friendly variant: the Spike GRU. As illustrated in Fig.3(middle), this module integrates binary spike signals $\mathcal{O}_t$ and membrane potentials $\mathcal{X}_t$, both outputs of the preceding LIF layer, into the gating mechanism. A trainable voltage threshold $\mathcal{V}_{th}$ determines spike firing as:

$$\mathcal{X}_t^{inter} = \mathcal{X}_t - \mathcal{V}_{th}, \quad \mathcal{Y}_t = \begin{cases} 1, & \text{if } \mathcal{X}_t^{inter} > 0 \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

The update gate $\mathcal{Z}_t$ is computed using convolution over both the previous spikes $\mathcal{O}_t$ and the current spikes $\mathcal{Y}_t$:

$$\mathcal{Z}_t = \sigma(\text{Conv}(\mathcal{O}_t) + \text{Conv}(\mathcal{Y}_t)), \tag{8}$$

The hidden state $\mathcal{H}_t$ is updated via:

$$\mathcal{H}_t = (1 - \mathcal{Z}_t) * \mathcal{H}_{t-1} + \mathcal{Z}_t * \text{Tanh}(\text{Conv}(\mathcal{Y}_t)). \tag{9}$$

By using binary spikes in place of continuous activations and reducing the number of convolution layers, the Spike GRU substantially lowers energy cost, while preserving the temporal recurrence and spatial reasoning required for event-based optical flow.

Rotation Estimation & SO(3) Refinement

To estimate the camera's global rotation from sparse events, we first extract a small number of high-activity points from the event voxel tensor. Specifically, we divide the sensor into spatial patches and select one salient point per patch to ensure spatial coverage and temporal saliency. These selected points are visualized in Fig. 4 as green stars. Using the SNN-predicted optical flow, each point is displaced to a new position (red stars), forming matched pairs in pixel space. The detailed point selection strategy is described in Tab.4.

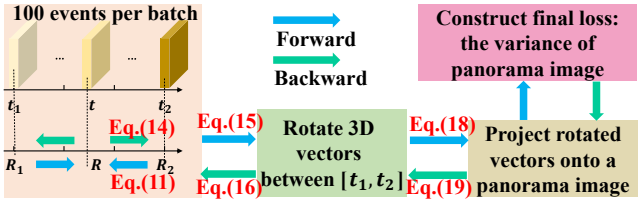


Figure 5: Overview of the $SO(3)$ refinement pipeline.

As shown as Fig.1(②,③), let (u, v) and (u', v') denote the original and displaced 2D locations, respectively. We back-project them to 3D camera space using the inverse projection π^{-1}, defined by the intrinsic parameters of the camera, obtaining 3D point pairs $(\mathbf{x}, \mathbf{x}') \in \mathbb{R}^3$. With these point correspondences, we estimate the relative rotation matrix $R \in SO(3)$ that minimizes the alignment error via:

$$R^* = \underset{R \in SO(3)}{\arg\min} \|\mathbf{X} - R\mathbf{X}'\|_F^2, \tag{10}$$

where $\mathbf{X}$ and $\mathbf{X}'$ denote matrices of matched 3D vectors.

Although this solution provides a closed-form estimate of the rotation, it is sensitive to noisy flow predictions or inaccurate correspondences. Therefore, we refine the result using a gradient-based optimization on $SO(3)$. By perturbing the estimated rotation along the Lie algebra and minimizing a geometric loss (e.g., projected sharpness or alignment error), we further improve the rotation accuracy in a principled and stable manner.

As shown in Fig.5, in our pipeline, the rotation matrices estimated from flow are discrete over time. To enable continuous optimization, we interpolate the rotation at time $t \in [t_1, t_2]$ by spherical interpolation between the two discrete estimates R_1 and R_2. Specifically, we define:

$$R = R_{21}^{\alpha} R_1 = (R_2 R_1^T)^{\alpha} R_1, \;\; \alpha = \frac{t - t_1}{t_2 - t_1} \in [0, 1], \quad (11)$$

where $R_{21} = R_2 R_1^T$ denotes the relative rotation and α controls the interpolation. To refine this interpolated rotation, we introduce small perturbations $\delta\phi_1$ and $\delta\phi_2 \in \mathbb{R}^3$ on the Lie algebra of $SO(3)$, leading to perturbed rotations:

$$R_1^{'} = \exp(\delta\phi_1^{\wedge}) R_1, \; R_2^{'} = \exp(\delta\phi_2^{\wedge}) R_2, \quad (12)$$

where $\wedge$ denotes the skew-symmetric operator that maps a vector to its corresponding element in $\mathfrak{so}(3)$, the Lie algebra of $SO(3)$. The inverse operation is denoted $\vee$. The perturbed interpolated rotation is $R^{'} = \exp(\delta\phi^{\wedge}) R$. After simplification (see Appendix for full derivation), the net perturbation on R becomes:

$$\delta\phi = (R_{21}^{\alpha} - \Gamma R_{21}) \delta\phi_1 + \Gamma \delta\phi_2, \quad (13)$$

where $\Gamma = \alpha J_{(\alpha\phi_{21})} J_{(\phi_{21})}^{-1}$, and $J_{(\phi_{21})}$ is the left Jacobian of $SO(3)$. Finally, we apply the refined interpolation result in downstream optimization by minimizing a contrast-based loss on the projected panorama image. Based on Eq.(13), the derivative of the Lie-algebra perturbation with respect to the input rotations is:

$$\frac{\partial \delta\phi}{\partial \delta\phi_1} = R_{21}^{\alpha} - \Gamma R_{21}, \;\; \frac{\partial \delta\phi}{\partial \delta\phi_2} = \Gamma. \quad (14)$$

Let $\mathbf{x} \in \mathbb{R}^3$ be a back-projected unit vector from the image plane. After rotation, the transformed vector becomes:

$$\mathbf{x}^{'} = \exp(\delta\phi^{\wedge}) R\mathbf{x} \approx R\mathbf{x} - R\mathbf{x}^{\wedge} \delta\phi, \quad (15)$$

thus yielding the gradient:

$$\frac{\partial \mathbf{x}^{'}}{\partial \delta\phi} = -R\mathbf{x}^{\wedge}. \quad (16)$$

We then project the rotated vector $\mathbf{x}^{'} = (x, y, z)^T$ onto a panorama image using equirectangular projection. Given the image width w, height h, and focal lengths f_x, f_y, the horizontal and vertical angles are:

$$\zeta = \arctan \frac{x}{z}, \;\; \theta = \arcsin \frac{y}{\|\mathbf{x}'\|}. \quad (17)$$

The projected pixel coordinate $\mathbf{p} = (u_p, v_p)^T$ on the panorama image is:

$$\mathbf{p} = \left(\frac{w}{2} + \zeta f_x, \; \frac{h}{2} + \theta f_y \right)^T. \quad (18)$$

The Jacobian of the projection with respect to the 3D vector $\mathbf{x}^{'}$ is:

$$\frac{\partial \mathbf{p}}{\partial \mathbf{x}'} = \begin{bmatrix} \frac{f_x z}{x^2+z^2} & 0 & -\frac{f_x x}{x^2+z^2} \\ -\frac{f_y xy}{r^2\sqrt{x^2+y^2}} & \frac{f_y \sqrt{x^2+z^2}}{r^2} & -\frac{f_y yz}{r^2\sqrt{x^2+z^2}} \end{bmatrix}, \quad (19)$$

where $r^2 = x^2 + y^2 + z^2$.

Finally, we define a loss based on the variance (sharpness) of the panorama image. By applying the chain rule through the above derivatives, we compute gradients with respect to the original rotation matrices R_1 and R_2 via their perturbations $\delta\phi_1$ and $\delta\phi_2$, allowing for end-to-end refinement of the global rotation estimation.

Experiments

To validate the effectiveness of our SNN model, we conduct experiments on three publicly available datasets. We evaluate optical flow prediction primarily on the DSEC dataset(Gehrig et al. 2021), which provides dense optical flow ground truth aligned with high-resolution event streams. To assess generalization, we include supplementary results on MVSEC(Zhu et al. 2018a) (see Appendix). For camera rotation estimation, we use the ECRot dataset(Guo and Gallego 2024), a recently introduced benchmark designed for evaluating event-based rotation tracking in real-world scenes. Notably, ECRot does not include ground-truth camera poses; therefore, we adopt the results from CMax-SLAM (Guo and Gallego 2024) as a reference trajectory for quantitative comparison.

We employ two standard metrics for optical flow evaluation: average Endpoint Error (EPE) and k-Pixel Error (kPE). The EPE is calculated as the spatial mean of the $\mathcal{L}_2$ distance between predicted flow $\mathcal{F}_{final}$ and the ground truth $\mathcal{F}_{gt}$:

$$\text{EPE} = \frac{1}{H \times W} \sum \|\mathcal{F}_{final}^{(i,j)} - \mathcal{F}_{gt}^{(i,j)}\|_2. \quad (20)$$

The kPE measures the percentage of pixels whose prediction error exceeds k pixels, providing a robustness indicator to outliers.

Optical Flow Comparison. We begin by evaluating optical flow prediction on the DSEC-Flow benchmark. Quantitative results are presented in Tab.1. Our SNN-based method outperforms all existing spiking models in both average endpoint error (EPE) and k-pixel error (1PE, 2PE, 3PE). In particular, the variant using a conventional ConvGRU (denoted by †) achieves an EPE of 0.85, significantly improving over previous works such as Adaptive SNN (ASNN)(Kosta and Roy 2023), EVSNN(Cuadrado et al. 2023), and SDformer (Tian and Andrade-Cetto 2024), which report EPEs of 1.62, 1.71, and 1.60, respectively.

Our full model, incorporating the proposed Spike GRU, yields an EPE of 0.87, only a slight increase over the ConvGRU variant, highlighting the effectiveness of our spiking recurrent design. Furthermore, when we disable membrane potential carryover across time segments (denoted by $*$), performance drops substantially to 1.97 EPE. This result underscores the importance of temporal continuity in SNNs for modeling long-term dynamics in event streams.

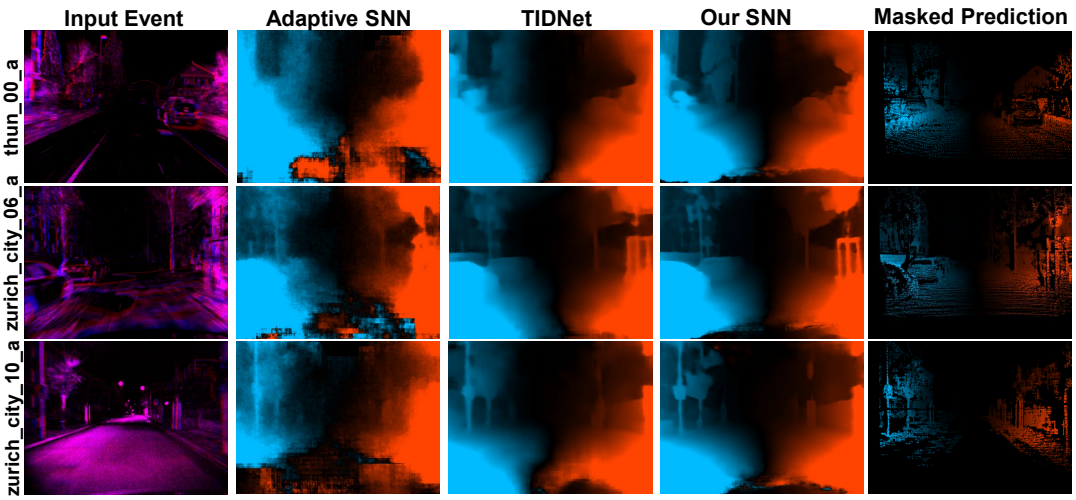


Figure 6: Qualitative results of event-based optical flow estimation on the DSEC-Flow validation set. Each sample is selected from a different sequence, with background scenes including daytime, dusk, and nighttime. The last column shows the masked prediction, obtained by applying the ground truth valid mask to the prediction.

	Method	EPE	1PE	2PE	3PE
ANN	EV-Flow	2.32	55.4	29.8	18.6
	ADM	0.89	12.5	6.2	5.6
	DCT	0.75	11.9	4.4	2.4
	TAM	2.33	-	-	17.8
	IDNet	**0.72**	**10.1**	**3.5**	**2.0**
	TIDNet	0.84	14.7	5.0	2.8
SNN	ASNN	1.62	19.2	8.4	4.5
	EVSNN	1.71	-	-	10.3
	SDformer	1.60	-	-	10.1
	Our$*$	1.97	18.5	10.6	8.5
	Our†	**0.85**	**10.4**	**5.2**	**2.7**
	Our	0.87	11.1	5.4	2.8

Table 1: Experimental results on the DSEC-Flow dataset. $*$ indicates a model variant without membrane potential carry-over between 10ms segments; † denotes the use of conventional ConvGRU instead of Spike GRU.

Compared to ANN-based methods, our SNN achieves competitive performance across all evaluation metrics. As shown in Tab. 1, although IDNet(Wu, Paredes-Vallés, and De Croon 2024) and DCT(Gehrig, Muglikar, and Scaramuzza 2024) obtain lower EPEs (0.72 and 0.75, respectively), our model reports comparable 1PE (11.1%) and 2PE (5.4%) values, outperforming ADM(Luo et al. 2023) (1PE: 12.5%) and TIDNet (1PE: 14.7%). The 3PE metric for our method (2.8%) matches that of TIDNet, further validating the spatial reliability of our prediction. Importantly, TIDNet shares the same overall network architecture as our model, making it a direct ANN baseline for comparison. Although our model is fully spike-based, it exhibits only a slight increase in EPE compared to TIDNet (0.87 vs. 0.84), while achieving comparable or better accuracy at higher error thresholds (1PE–3PE). This highlights the effectiveness of our direct-training strategy for SNNs, demonstrating that competitive performance can be achieved even under stringent spiking constraints.

Figure 6 presents qualitative results across a range of environmental conditions including daytime, dusk, and nighttime. Our model consistently produces sharper and more spatially detailed flow fields compared to existing SNN approaches such as ASNN, and even outperforms TIDNet in terms of visual clarity in several cases. The masked predictions in the last column, where the valid regions are highlighted using ground truth masks, further illustrate the precision of our estimates. Additional results on the official DSEC test set are provided in the Appendix.

Efficiency Analysis. To assess energy efficiency, we estimate power consumption following the methodology proposed in prior works (Yao et al. 2024; Zhou et al. 2022). For ANN models, energy is computed as the total number of multiply-accumulate operations (FLOPs) multiplied by the energy per MAC operation (E_{MAC}). In SNNs, where information is transmitted through binary spikes ($S \in \{0, +1\}$), convolution involves only additions. Therefore, the energy is approximated as $\text{FLOPs} \times E_{\text{AC}} \times T \times F_r$, where E_{AC} is the energy per addition, T is the number of time steps, and F_r is the average firing rate. We use the widely adopted estimates from (Horowitz 2014), with $E_{\text{MAC}} = 4.6$ pJ and $E_{\text{AC}} = 0.9$ pJ. In our design, the Input Encoder and Spike GRU operate over 10 time steps, while the Decoder and Flow Encoder use 1 step. FLOPs are computed using $C_{\text{in}} \times C_{\text{out}} \times H \times W \times k^2$ per convolutional layer.

We evaluate the computational efficiency of our method in terms of parameter count, floating-point operations (FLOPs), and estimated energy consumption. As summarized in Tab.2, our SNN model is compared against both ANN and SNN baselines.

Among ANN-based methods, IDNet achieves the best accuracy (EPE = 0.72), but incurs extremely high computa-

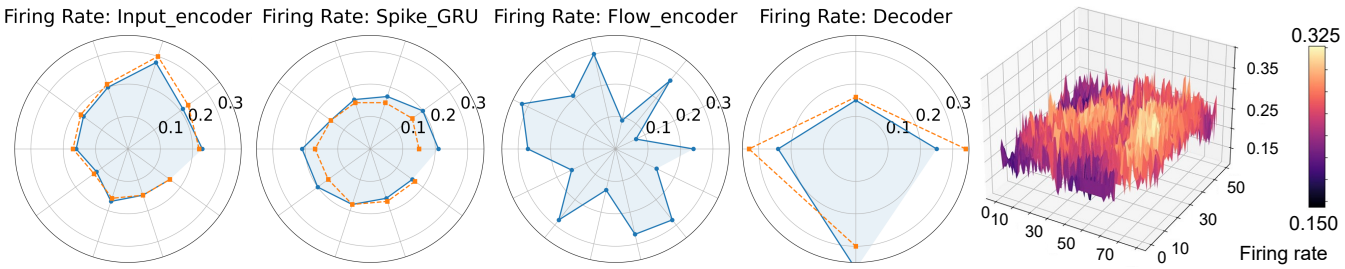


Figure 7: Visualization of firing rates across different modules in our SNN model. The radar charts show the average firing rates of each layer within the Input Encoder, Spike GRU, Flow Encoder, and Decoder modules, respectively. The 3D surface plot on the right illustrates the spatial firing rate distribution of the first convolutional layer in the Input Encoder.

	Model	EPE	Param.(M)	FLOPs(G)	$E_{Total}(pJ)$	Improve
ANN	EV-Flow	2.32	14.14	33.09	152.21	1×
	IDNet	0.72	1.67	222.46	1023.31	0.15×
	TIDNet	0.84	1.88	29.41	135.30	1.12×
SNN	Adaptive-SNN	1.62	13.04	95.47	8.84	17.22×
	SDformer	1.14	54.92	42.63	36.83	4.13×
	Our SNN	0.87	1.58	18.90	3.55	42.88×

Table 2: A comparison of computational complexity, parameter count, and power consumption between our proposed SNN model and other baseline models.

tional cost (222.46G FLOPs and 1023.31 pJ per inference). TIDNet, which shares the same architecture as our model, offers a more efficient alternative with 29.41G FLOPs and 135.30 pJ, while maintaining competitive accuracy (EPE = 0.84). In contrast, our proposed SNN model achieves a comparable EPE of 0.87 while significantly reducing the FLOPs to 18.90G and estimated energy consumption to just 3.55 pJ—representing a $42.88\times$ reduction compared to EV-FlowNet (Zhu et al. 2018b), and a $38\times$ improvement over TIDNet. These benefits are achieved with a similar number of parameters (1.58M vs. 1.88M).

Compared to other SNN-based approaches, our model strikes the best balance between accuracy and efficiency. Adaptive-SNN and SDformer consume more energy (8.84 pJ and 36.83 pJ, respectively) and yield higher EPEs (1.62 and 1.14). Notably, our model achieves both the lowest EPE and the lowest energy consumption among all SNNs evaluated.

Rotation Estimation. Fig. 8 shows qualitative comparisons of panoramic images generated using the estimated rotation trajectories from different methods. The reference image from CMax-SLAM (a) serves as the baseline, which provides visually consistent alignment across the scene. Compared to this, the image constructed using TIDNet-predicted optical flow (b) exhibits moderate distortions and blur, particularly along structural boundaries such as building edges. Our directly trained SNN model (c) achieves similar spatial coverage but exhibits mild distortions and reduced sharpness, likely due to the limited temporal precision and accumulated error in the recurrent spiking representation. After applying gradient-based $SO(3)$ refinement to our SNN output (d), the resulting panorama becomes significantly sharper, with clearly delineated lines and textures across the entire field of view. This refined image closely matches the geometric layout of CMax-SLAM, and even surpasses it in terms of visual clarity. The improvement is partially attributed to the application of gamma correction ($\gamma = 0.5$), which enhances contrast in the projected images.

Quantitative Rotation Comparison. We evaluate the accuracy of camera rotation estimation on multiple sequences from the ECRot dataset. Since no ground-truth poses are provided, we use rotation trajectories from CMax-SLAM as reference. The mean rotation error per frame is reported in Tab.3, measured by the geodesic distance between rotation matrices, computed as $\arccos\left(\frac{\mathrm{Tr}(R_1^\top R_2)-1}{2}\right)$. The reported mean error is averaged across all frames using the formula: Mean Error $= \frac{\text{Total Error}}{\text{Number of Frames}}$, where each frame corresponds to a 0.1s interval.

Despite not relying on inertial measurements or external pose supervision, our method achieves promising results. Notably, on full-rotation sequences such as Main Building (360°) and Brandenburg Gate (360°), we observe mean errors of 0.386 and 0.274 radians, respectively. On smaller-angle sequences like Victory Column (90°), the error reduces further to 0.193, indicating stable short-term tracking performance. $SO(3)$ refinement further reduces errors to 0.110, 0.084, and 0.023 radians, confirming its benefit for accurate, event-only rotation estimation.

In addition to accuracy, we also report runtime in Tab.3. All experiments were conducted on a laptop with an Intel Core i5-10400 CPU @ 2.90GHz. Our SNN can predict dense optical flow and estimate rotation with very low latency, demonstrating strong potential for real-time deployment. Although the optional $SO(3)$ refinement is slower, it is decoupled from the core SNN pipeline and can be omitted

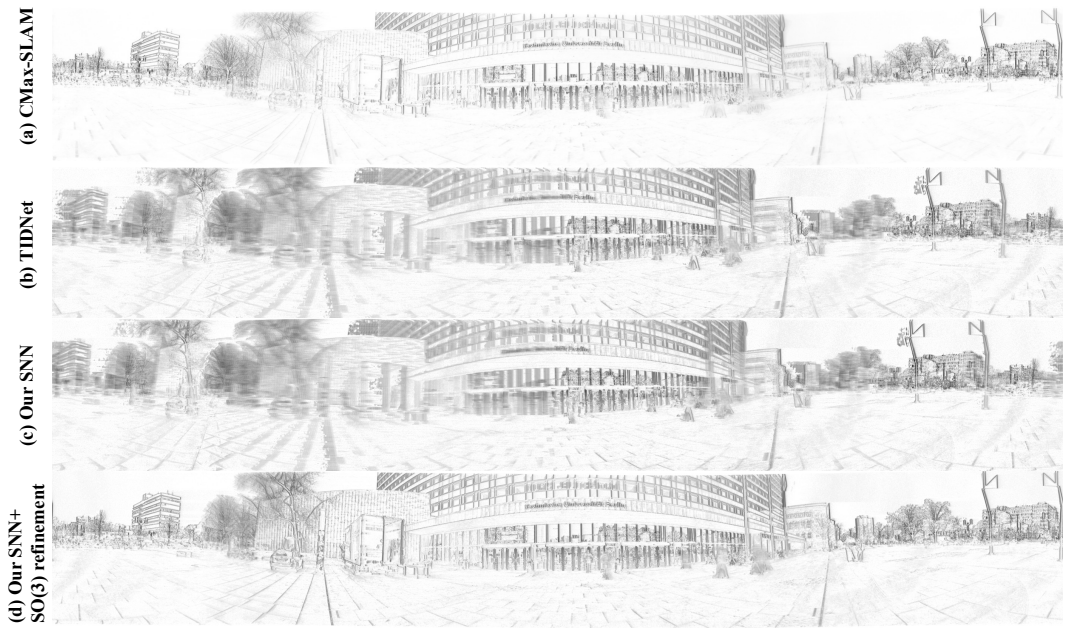


Figure 8: Qualitative comparison of panoramic images generated from estimated rotation trajectories. Our method with $SO(3)$ refinement recovers a clearer and more geometrically consistent panorama, surpassing TIDNet and matching the visual quality of CMax-SLAM.

Sequence	Angle	Duration	Events	Mean Error	Runtime (s)			
	$(^\circ)$	(s)	$(\times 10^6)$	SNN / +$SO(3)$	CMax-SLAM	Our SNN	+$SO(3)$ Refine	Pano. Time
Main Bld.	360	8.5	116.2	0.386 / 0.110	779	54	267	43
Brand. Gt.	360	8.0	97.4	0.274 / 0.084	1143	47	452	39
Mar Bld.	≈90	4.0	33.8	0.202 / 0.061	358	12	131	18
Vict. Col.	90	10.0	4.9	0.193 / 0.023	745	14	202	15

Table 3: Rotation estimation accuracy and runtime comparison on representative ECRot sequences. We report mean error per frame and total runtime for CMax-SLAM, our SNN-based estimation, optional $SO(3)$ refinement, and panorama construction.

Sequence	8	16	24	32
Main Bld.	0.420	0.386	0.422	0.422
Brand. Gt.	0.381	0.394	0.394	0.394
Vict. Col.	0.193	0.193	0.193	0.193

Table 4: Rotation estimation accuracy of our SNN model under different numbers of extracted event keypoints.

in latency-critical settings. Even when including refinement and panorama construction, our total runtime remains significantly lower than that of CMax-SLAM, highlighting the efficiency of our lightweight, event-driven design.

To investigate the effect of keypoint density on rotation estimation, we evaluate the model with varying numbers of extracted event points, as summarized in Tab.4. We adopt a spatially-aware strategy to select event points: each input voxel tensor is divided into 8×8 patches, and we compute the total event activation in each patch. The top-scoring patches are selected based on their total activity (a form of local top-k filtering), followed by local non-maximum suppression (NMS) to extract salient points.

As shown in Tab. 4, using 16 points (i.e., selecting the top 16 most active patches with 1 point each) yields the best trade-off between stability and robustness across different sequences. Increasing the number of points beyond this does not significantly improve accuracy, and in some cases, slightly degrades performance likely due to the inclusion of low-saliency or noisy regions.

Conclusion

We present a fully spiking neural network for event-based optical flow estimation using a Spike GRU and membrane potential carryover. Our approach achieves state-of-the-art accuracy among SNNs with greatly reduced energy consumption. We also estimate camera rotations and construct panoramic event images purely from event streams. An unsupervised $SO(3)$ refinement further improves geometric consistency without external sensors. Experiments show our event-driven pipeline matches complex systems like CMax-SLAM while being simpler and more efficient.

Acknowledgments

This work was supported by the National Key R&D Program of China (No. 2025YFG0100400), the grants from Key R&D Program of Zhejiang (No. 2022C01048), and Key Program of National Natural Science Foundation of China (62334014).

References

Akopyan, F.; Sawada, J.; Cassidy, A.; Alvarez-Icaza, R.; Arthur, J.; Merolla, P.; Imam, N.; Nakamura, Y.; Datta, P.; Nam, G.-J.; et al. 2015. 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.

Ballas, N.; Yao, L.; Pal, C.; and Courville, A. 2015. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432*.

Bouwmeester, R. J.; Paredes-Vallés, F.; and De Croon, G. C. 2023. Nanoflownet: Real-time dense optical flow on a nano quadcopter. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 1996–2003. IEEE.

Cuadrado, J.; Rançon, U.; Cottereau, B. R.; Barranco, F.; and Masquelier, T. 2023. Optical flow estimation from event-based cameras and spiking neural networks. *Frontiers in Neuroscience*, 17: 1160034.

Davies, M.; Srinivasa, N.; Lin, T.-H.; Chinya, G.; Cao, Y.; Choday, S. H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1): 82–99.

Gallego, G.; Delbrück, T.; Orchard, G.; Bartolozzi, C.; Taba, B.; Censi, A.; Leutenegger, S.; Davison, A. J.; Conradt, J.; Daniilidis, K.; et al. 2020. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1): 154–180.

Gehrig, M.; Aarents, W.; Gehrig, D.; and Scaramuzza, D. 2021. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6(3): 4947–4954.

Gehrig, M.; Muglikar, M.; and Scaramuzza, D. 2024. Dense continuous-time optical flow from event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Gerstner, W.; Kistler, W. M.; Naud, R.; and Paninski, L. 2014. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press.

Guo, S.; and Gallego, G. 2024. CMax-SLAM: Event-based Rotational-Motion Bundle Adjustment and SLAM System using Contrast Maximization. *IEEE Transactions on Robotics*, 1–20.

Horowitz, M. 2014. 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)*, 10–14. IEEE.

Kosta, A. K.; and Roy, K. 2023. 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)*, 6021–6027. IEEE.

Lipson, L.; Teed, Z.; and Deng, J. 2024. Deep patch visual slam. In *European Conference on Computer Vision*, 424–440. Springer.

Luo, X.; Luo, K.; Luo, A.; Wang, Z.; Tan, P.; and Liu, S. 2023. Learning optical flow from event camera with rendered dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9847–9857.

Ma, D.; Jin, X.; Sun, S.; Li, Y.; Wu, X.; Hu, Y.; Yang, F.; Tang, H.; Zhu, X.; Lin, P.; et al. 2024. Darwin3: a large-scale neuromorphic chip with a novel ISA and on-chip learning. *National Science Review*, 11(5): nwae102.

Pellerito, R.; Cannici, M.; Gehrig, D.; Belhadj, J.; Dubois-Matra, O.; Casasco, M.; and Scaramuzza, D. 2024. Deep visual odometry with events and frames. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 8966–8973. IEEE.

Tian, Y.; and Andrade-Cetto, J. 2024. SDformerFlow: Spatiotemporal swin spikeformer for event-based optical flow estimation. *arXiv preprint arXiv:2409.04082*.

Wu, Y.; Paredes-Vallés, F.; and De Croon, G. C. 2024. Lightweight event-based optical flow estimation via iterative deblurring. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 14708–14715. IEEE.

Yao, M.; Hu, J.; Zhou, Z.; Yuan, L.; Tian, Y.; Xu, B.; and Li, G. 2024. Spike-driven transformer. *Advances in neural information processing systems*, 36.

Zhou, Z.; Zhu, Y.; He, C.; Wang, Y.; Yan, S.; Tian, Y.; and Yuan, L. 2022. Spikformer: When spiking neural network meets transformer. *arXiv preprint arXiv:2209.15425*.

Zhu, A. Z.; Thakur, D.; Özaslan, T.; Pfrommer, B.; Kumar, V.; and Daniilidis, K. 2018a. The multivehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robotics and Automation Letters*, 3(3): 2032–2039.

Zhu, A. Z.; Yuan, L.; Chaney, K.; and Daniilidis, K. 2018b. EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*.