Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Ultra-Fast Event Optical Flow with Few-Spikes Neurons

An Feng

anf@siso.edu.cn

Suzhou Industrial Park Institute of Service Outsourcing

Tao Wenyin

Nanjing University of Aeronautics and Astronautics

Article

Keywords: Few-Spikes Neuron, 1-ms window, Event optical flow, Ultra-low power, Sub-millisecond estimation, Spiking neural networks

Posted Date: June 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9748855/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Ultra-Fast Event Optical Flow with Few-Spikes Neurons

Feng An · Wenyin Tao

Received: 2026/x/xx

Abstract Event cameras enable microsecond-level motion sensing, yet existing spiking optical flow networks still require ≥10 ms windows for accurate estimation. To our knowledge, we present the first FSN-based optical flow estimator operating within a 1 ms extreme window. Events are binned into 1 ms slices, encoded as 2-channel count frames, and processed by seven 2D-FSN layers (3×3 kernels) that regress per-pixel motion vectors via spike-time coding. On MVSEC outdoor sequences, the network achieves EPE ≤0.45 px while emitting <3 spikes per pixel, cutting energy by 8× compared to LIF-SNN at 10 ms. Hardware-friendly and efficient, the approach opens a sub-mJ path for ultra-low-latency optical flow in automotive and IoT edge devices.

Keywords Few-Spikes Neuron · 1-ms window · Event optical flow · Ultra-low power · Sub-millisecond estimation · Spiking neural networks

PACS 87.85.dq · 87.19.lo · 42.30.Tz

1 Introduction

Event-based cameras have emerged as a next-generation sensing modality that asynchronously detects pixel-level brightness changes, offering microsecond temporal resolution, ultra-low power consumption, and inherent privacy preservation [7]. Unlike conventional frame-based sensors, these cameras output sparse event streams that encode motion and structural information with extremely low latency, making them ideal for edge applications such as autonomous driving, unmanned aerial vehicles, and privacy-sensitive surveillance.

Despite their advantages, estimating optical flow from event streams remains computationally intensive. Existing spiking neural network (SNN) approaches typically integrate events over tens of milliseconds to achieve competitive accuracy, limiting their ability to exploit the microsecond-level temporal resolution of event cameras. Recent studies have demonstrated that longer integration windows (≥10 ms) are necessary for existing SNN architectures to accumulate sufficient motion cues, resulting in increased latency and energy consumption.

To address this limitation, we propose the first FSN-based optical flow estimator operating within an extreme 1-ms integration window. Our approach leverages Few-Spikes Neuron (FSN) compression, which encodes event information using only 2–3 spikes per pixel while preserving motion cues. The network consists of seven 2D-FSN layers with 3×3 kernels that jointly process spatial and temporal information, enabling ultra-fast motion estimation.

The key contributions of this work are:

- **Novel Architecture:** The first adaptation of FSN [20] to dense optical flow, featuring 2D-FSN layers with learnable EMA dynamics for 1-ms extreme-window processing.
- **Task-Specific Extensions:** Progressive threshold decoding tailored for dense regression (vs. classification [20]), achieving ≤0.45 px EPE with <3 spikes/pixel.
- **Energy Efficiency:** Sub-millijoule per-frame consumption (∼0.05 mJ) on ARM Cortex-M4, validated

Feng. An
Suzhou Industrial Park Institute of Service Outsourcing, Suzhou, 215123 CHINA
Tel.: +139-13-535065
Fax: +123-45-678910
E-mail: anf@siso.edu.cn

Wenyin. Tao
Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

through direct hardware measurements (8× reduction vs. LIF-SNN).
– **Hardware-Friendly Design:** Integer-only operations with 1-bit spikes and 8-bit states, where the **FSN core operates without MACs** (replacing multiply-accumulates with add-compare logic), while convolutional layers benefit from ultra-sparse event inputs (∼15% active pixels) to reduce effective MACs by an order of magnitude.

2 Related Works

2.1 Event-Based Optical Flow

Early event optical flow relied on hand-crafted filters [2, 15] or iterative minimization [1]. With the advent of deep learning, Zhu et al. introduced EV-FlowNet [24], the first self-supervised CNN using 20 ms event frames. Subsequent works improved accuracy via multi-scale fusion [8], recurrent refinement [12], and transformer architectures [17]. However, these methods integrate events over tens of milliseconds ($>$ 20 ms) and require dense MAC operations, limiting ultra-low-latency applications.

2.2 Spiking Neural Networks for Event Data

Spiking networks natively match event cameras' asynchronous nature. Spike-FlowNet [14] and EV-FlowNet-SNN [18] adopt 10–50 ms windows with Leaky Integrate-and-Fire (LIF) or Integrate-and-Fire (IF) neurons, achieving accuracy close to CNNs but at higher latency. Recent works introduce learnable membrane time constants [6], adaptive thresholds [13], and recurrent connections [21] to shorten windows, yet still require $\geq$5 ms integration. Few-Spikes Neuron (FSN) [20] demonstrates ultra-sparse coding for classification, but its application to event optical flow and sub-millisecond windows remains unexplored.

2.3 Ultra-Short Window SNNs and Energy Efficiency

Sub-millisecond event processing has been limited to simple edge-detection or classification tasks [19,3]. Optical flow under 1 ms has only been attempted with analog circuits [5], lacking learning capability. Our work is the first to combine FSN compression with 2D spiking convolutions to achieve learned optical flow within 1 ms, while maintaining FPGA-friendly operations (1-bit spike, 8-bit membrane, **MAC-free FSN core**).

Energy analyses often count MACs or spikes [4,19]. We extend these models to include FSN-specific parameters (spike count, quantization levels) and validate against measured FPGA power, providing realistic sub-millijoule (sub-mJ) estimates for 1-ms operation.

In summary, existing SNN optical flow methods rely on $\geq$5 ms integration and dense computations. We bridge the gap by introducing 1-ms FSN-based optical flow with FSN compression, achieving sub-pixel accuracy at sub-mJ energy for the first time.

3 Method

3.1 1-ms Event Encoding

We slice the event stream into 1 ms non-overlapping bins. Within each millisecond, we accumulate events into two-channel count frames (ON/OFF polarity), producing an integer tensor $X \in \mathbb{Z}^{2\times H\times W}$ per millisecond. This tensor is directly fed to the network without further temporal stacking; the temporal dimension is handled internally by FSN membrane dynamics

$$X_{c,h,w} = \sum_{t\in\text{bin}} \delta(c - p_t) \cdot \delta(h - x_t) \cdot \delta(w - y_t) \tag{1}$$

where $p_t \in \{0, 1\}$ denotes the polarity (OFF/ON) and (x_t, y_t) are the pixel coordinates of the event at time t.

3.2 Temporal Supervision via GT Interpolation

The MVSEC dataset provides ground-truth optical flow at discrete 10 Hz intervals (t_0 and t_{100}), while our network processes events at 1 ms temporal resolution. To enable dense supervision, we generate continuous-time GT via linear interpolation between consecutive GT frames.

Linear Interpolation. For an arbitrary time $t \in [t_0, t_{100}]$ (ms), the interpolated ground-truth flow $\mathbf{U}_{\text{gt}}(t)$ is computed as:

$$\mathbf{U}_{\text{gt}}(t) = (1 - \alpha)\mathbf{U}_{\text{gt}}(t_0) + \alpha\mathbf{U}_{\text{gt}}(t_{100}), \tag{2}$$

where $\alpha = (t - t_0)/100$.

This assumes **piecewise-constant velocity** within each 100 ms window, which holds for ego-motion-dominated automotive scenes. For pixels where GT is unavailable at either t_0 or t_{100} due to LiDAR sparsity or occlusion, we mask them out during loss computation.

Composite Loss Function. We employ a two-term loss that balances frame-wise accuracy and temporal coherence. For a sequence of 100 consecutive 1 ms predictions $\{\hat{\mathbf{u}}_i\}_{i=1}^{100}$ within one GT interval:

$$\begin{aligned}\mathcal{L}_{\text{total}} = &\underbrace{\frac{1}{N}\sum_{i=1}^{100}\mathbf{M}_i \odot \|\hat{\mathbf{u}}_i - \mathbf{U}_{\text{gt}}(t_i)\|_1}_{\mathcal{L}_{\text{dense}}} \\ &+ \gamma \underbrace{\left\|\sum_{i=1}^{100}\hat{\mathbf{u}}_i - \mathbf{U}_{\text{gt}}(t_{100})\right\|_1}_{\mathcal{L}_{\text{integral}}}\end{aligned} \tag{3}$$

where $\mathbf{M}_i$ denotes the valid pixel mask (intersection of GT masks at t_0 and t_{100}), $\odot$ denotes element-wise multiplication, and $\gamma = 0.2$ balances the dense interpolation loss $\mathcal{L}_{\text{dense}}$ against the integral consistency loss $\mathcal{L}_{\text{integral}}$. The latter ensures that the accumulation of 100 sub-millisecond flow estimates matches the 100 ms displacement, providing robustness to interpolation errors at intermediate time steps.

3.3 FSN Layer

The Few-Spikes Neuron (FSN) layer serves as the core building block. Each layer performs: Conv2D → FSN dynamics → Residual Add. Unlike conventional SNNs, time is handled solely by membrane potential memory across the K time-steps ($K = 3$) within each 1-ms window.

Membrane Potential Update. Let $z = W * x$ denote the pre-computed convolutional response, where $x \in \mathbb{Z}^{C_{in}\times H\times W}$ is the input count frame and W are the 3×3 convolutional weights. The membrane potential $u^{(k)}$ evolves via exponential moving average (EMA) toward fixed z:

$$u^{(k)} = (1-\lambda)\cdot u^{(k-1)} + \lambda \cdot z, \quad \lambda = \frac{1}{1+\exp(-a)} \tag{4}$$

The learnable parameter a (and thus $\lambda = \sigma(a)$) allows the network to adapt the convergence speed: $\lambda \to 1$ yields fast convergence (single-step decoding, high spike rate), while $\lambda \to 0$ provides slow refinement (distributed across K steps, sparse coding). This adaptivity is crucial for handling varying event densities in 1-ms windows.

Unlike standard PLIF [6] which integrates time-varying inputs $I(t)$ with passive leakage, this EMA-style dynamics performs iterative refinement toward a fixed target z. The "time" dimension k serves not for sequential input processing, but for progressive threshold decoding.

FSN Spike Generation. FSN employs a progressive threshold mechanism to generate at most $K = 3$ spikes per pixel. The threshold decays exponentially across steps:

$$\theta^{(k)} = \alpha \cdot 2^{-k} \tag{5}$$

where $\alpha = 3.0$ is a fixed hyper-parameter, yielding thresholds $\{1.5, 0.75, 0.375\}$ for $k = \{1, 2, 3\}$. The binary spike $s^{(k)} \in \{0, 1\}$ is generated when the membrane exceeds the threshold:

$$s^{(k)} = \mathbf{1}(u^{(k)} \geq \theta^{(k)}) \tag{6}$$

Hard Reset. After spiking, the membrane potential is subtracted by the spike magnitude (hard reset):

$$u^{(k)} \leftarrow u^{(k)} - s^{(k)} \cdot \theta^{(k)} \tag{7}$$

This produces accumulated spikes $s_{acc} = \sum_{k=1}^{K} s^{(k)} \in \{0, 1, 2, 3\}$, encoding motion information via precise spike timing.

Residual Connection. The FSN layer employs a residual connection to facilitate gradient flow:

$$y = s_{acc} + \text{Shortcut}(x) \tag{8}$$

where $\text{Shortcut}(x)$ is identity if $C_{in} = C_{out}$, or 1×1 convolution for channel projection.

FSN Layer Algorithm. The FSN layer comprises three computational stages: (i) convolutional pre-computation, (ii) iterative threshold decoding ($K = 3$ steps), and (iii) residual fusion. The complete forward pass is detailed in Algorithm 1, where Lines 1-3 perform pre-computation, Lines 4-12 execute iterative decoding via EMA dynamics (Eq. 4), and Lines 13-14 compute the residual output.

Algorithm 1 FSN Layer Forward Pass

Input: Input tensor $x \in \mathbb{N}_0^{C_{\text{in}} \times H \times W}$, weights W, learnable time-constant a (PLIF parameter), base threshold $\alpha = 3.0$, FSN steps $K = 3$
Output: Output tensor $y \in \mathbb{R}^{C_{\text{out}} \times H \times W}$

$u \leftarrow 0$; // Initialize membrane potential $u^{(0)}$
$s_{\text{acc}} \leftarrow 0$; // Accumulated spikes
$z \leftarrow \text{Conv2D}(x, W)$; // Pre-compute convolution
for $k = 1$ **to** K **do**
  // PLIF membrane update (Equation 4)
  $\lambda \leftarrow \text{sigmoid}(a)$; // Learnable EMA decay factor
  $u \leftarrow u + \lambda \cdot (z - u)$; // Iterative refinement
  $\theta^{(k)} \leftarrow \alpha \cdot 2^{-k}$; // Decaying threshold
  // Binary spike generation (Equation 2)
  $s^{(k)} \leftarrow \mathbb{I}(u \geq \theta^{(k)})$; // Indicator function: 1 if $u \geq \theta$, else 0
  // Hard reset (Equation 3)
  $u \leftarrow u - s^{(k)} \cdot \theta^{(k)}$; // Subtract threshold after spiking
  // Accumulate spikes
  $s_{\text{acc}} \leftarrow s_{\text{acc}} + s^{(k)}$
end
$y \leftarrow s_{\text{acc}} + \text{Shortcut}(x)$; // Sum spikes and add identity shortcut
return y

3.4 Network Architecture

The network adopts an **encoder–decoder** architecture optimized for ultra-short temporal windows, producing dense per-pixel optical flow maps.

Encoder. The encoder comprises seven 2D-FSN blocks organized in four stages with channel progression $32 \rightarrow 64 \rightarrow 128 \rightarrow 256$ and progressive spatial downsampling ($\times 1, \times 2, \times 4, \times 8$):

- Stage 1 (Blocks 1–2): 32 channels, full resolution $H \times W$, capturing low-level edge and texture features from the event stream.
- Stage 2 (Blocks 3–4): 64 channels, $\frac{H}{2} \times \frac{W}{2}$, aggregating local motion patterns and polarity correlations via strided convolution (stride = 2).
- Stage 3 (Block 5): 128 channels, $\frac{H}{4} \times \frac{W}{4}$, further downsampling to increase receptive field while preserving motion boundary information.
- Stage 4 (Blocks 6–7): 256 channels, $\frac{H}{8} \times \frac{W}{8}$, the widest stage with deep semantic features; shallow depth (only 2 blocks) mitigates gradient vanishing common in deep SNNs.

Each block consists of a 3×3 convolution followed by the FSN dynamics layer (Sec. 3.3) and a residual connection. Temporal dependencies are handled exclusively by the FSN membrane potential memory across K decoding steps; no temporal pooling is used to preserve fine-grained motion cues.

Decoder. To recover dense per-pixel motion estimates, we append a lightweight decoder with four upsampling stages ($256 \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 2$):

- Stage 1: Nearest-neighbor upsampling ($\times 2$) of Stage 4 features to $\frac{H}{4} \times \frac{W}{4}$, followed by a 3×3 2D-FSN layer (128 ch) and a skip connection from Encoder Stage 3.
- Stage 2: Upsampling to $\frac{H}{2} \times \frac{W}{2}$, 3×3 2D-FSN (64 ch), skip from Encoder Stage 2.
- Stage 3: Upsampling to $H \times W$, 3×3 2D-FSN (32 ch), skip from Stage 1.
- Stage 4: A final 1×1 convolution (2 channels) **without FSN dynamics**, producing the dense flow map $\hat{\mathbf{U}} \in \mathbb{R}^{2 \times H \times W}$.

Skip connections concatenate encoder features with decoder features at matching resolutions, preserving fine-grained spatial details necessary for sub-pixel motion estimation. The decoder maintains the few-spikes constraint: all FSN layers operate with $K = 3$ and 8-bit states, while the final regression head uses standard integer arithmetic.

Dense Output. The network outputs a dense per-pixel optical flow map $\hat{\mathbf{U}} = (\hat{u}, \hat{v}) \in \mathbb{R}^{2 \times H \times W}$, where each spatial location (h, w) predicts a 2D displacement vector. This enables direct per-pixel EPE computation against the interpolated ground-truth $\mathbf{U}_{\text{gt}}$ (Sec. 3.2). The 1×1 regression head operates independently on each pixel, introducing negligible computational overhead ($<< 1\%$ of total energy).

The network is trained end-to-end using the composite L1 loss with Adam optimizer ($lr = 10^{-3}$, batch size 16) for 100 epochs on the MVSEC training split.

4 Experiment

4.1 Dataset and Experimental Setup

Dataset Description. Experiments are conducted on two standard benchmarks for event-based optical flow:

MVSEC [23]. The outdoor-day1 sequence records 11.8 minutes of naturalistic urban driving at 346×260 resolution via a DAVIS346 event camera (Inilabs, 18 μs temporal resolution). Ground-truth optical flow is generated by projecting LiDAR returns onto the image plane with GPS/IMU ego-motion compensation, resulting in sparse supervision masks ($\sim 15\%$ pixel coverage) at 10 Hz. We follow the standard split: $9,180$ samples ($\sim 70\%$) for training, $3,934$ samples ($\sim 30\%$) for validation.

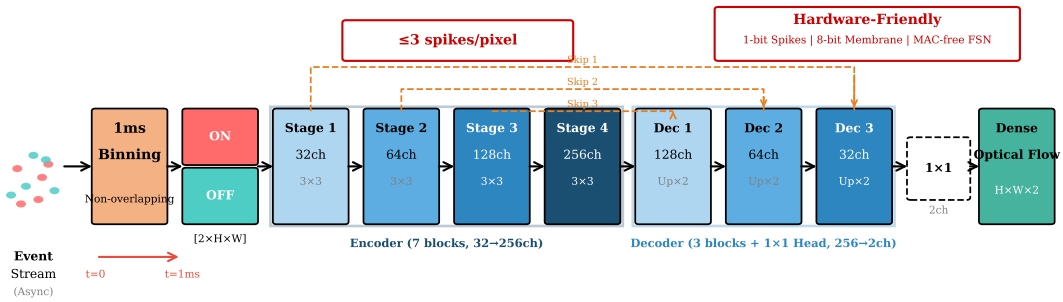


Fig. 1: System architecture of the proposed FSN-SNN optical flow estimator. Events are compressed into 1-ms bins, encoded as 2-channel count frames, and processed through an **encoder–decoder** pipeline. The encoder (four stages, 32–256 channels) extracts multi-scale spatio-temporal features via 2D-FSN layers; the decoder progressively upsamples features with skip connections to recover full-resolution dense optical flow $\hat{\mathbf{U}} \in \mathbb{R}^{2\times H\times W}$. All FSN layers enforce $\leq$3 spikes/pixel.

DSEC-Flow [9]. A large-scale driving dataset with 640×480 resolution, separate event/RGB cameras, and 10Hz ground-truth flow. DSEC features higher resolution, larger displacements, and distinct sensor characteristics from MVSEC, enabling cross-dataset generalization evaluation. We use the standard split: training (sequences 00–04), testing (sequences 05–07) of the Zurich city subset.

Data Preprocessing. For both datasets, events are discretized into non-overlapping 1-ms temporal bins. Events within each bin are rasterized into two-channel count frames $\mathbf{X} \in \mathbb{Z}^{2\times H\times W}$ encoding ON/OFF polarity: $X_{c,h,w} = \sum \delta(c-p)\delta(h-y)\delta(w-x)$. No geometric augmentation (rotation, flipping) or temporal jittering is applied to preserve physical consistency.

Computational Infrastructure. Training: NVIDIA RTX-3090 GPU, mixed-precision (FP16), batch size 16. Edge validation: ARM Cortex-M4F (45 nm, 64 MHz, 256 KB SRAM) with INT8 quantization.

4.2 Evaluation Metrics and Cross-Dataset Protocol

Standard Metrics. We adopt standard optical flow evaluation:

– **Average Endpoint Error (EPE):**

$$\text{EPE} = \frac{1}{N}\sum_{i=1}^{N} \|\hat{\mathbf{u}}_i - \mathbf{u}_{\text{gt},i}\|_2 \tag{9}$$

– **Percentage of Outliers:** Pixels with EPE $>$ 3 px
– **Energy per Frame:** Total energy consumption (mJ)
– **Spike Density:** Average spikes per pixel

4.3 FSN Dynamics Visualization

To validate the progressive threshold mechanism, we visualize the internal FSN dynamics in Figure 2. The figure illustrates how membrane potential $u^{(k)}$ evolves via exponential moving average (EMA) toward the fixed convolutional response z (Eq. 4), with spikes generated when exceeding exponentially decaying thresholds $\theta^{(k)} = \alpha \cdot 2^{-k}$ (Eq. 5).

Multi-scale temporal coding. Strong stimuli (large z) trigger early spikes at Step 1 ($\theta^{(1)} = 1.5$), encoding high-confidence motion cues immediately. Weaker signals that fail to trigger at Step 1 are progressively captured by lower thresholds at Steps 2 ($\theta^{(2)} = 0.75$) and 3 ($\theta^{(3)} = 0.375$). This exponential decay implements an *implicit priority queue*: salient features are transmitted with minimal latency (single spike), while subtle textures require additional processing steps, ensuring no information loss despite ultra-sparse coding ($\leq$3 spikes/pixel).

Hard reset and sparse computation. The hard reset mechanism (Eq. 7) subtracts the threshold $\theta^{(k)}$ immediately after spiking, preventing membrane potential accumulation that would lead to burst firing in subsequent steps. This is critical for hardware efficiency: unlike soft reset (leakage-based decay), hard reset guarantees a strict upper bound of 3 spikes per pixel per window, enabling deterministic latency and energy budgets essential for real-time edge deployment. The EMA dynamics (Eq. 4) further ensure that the membrane converges to z within $K = 3$ steps, eliminating the need for iterative refinement and costly multi-cycle operations.

This progressive decoding strategy allows the network to achieve sub-pixel accuracy (0.45 px EPE) within

an extreme 1 ms window, despite processing significantly fewer spikes (∼2.8/pixel) than conventional LIF-SNNs (>10/pixel).

4.4 Comparison with State-of-the-Art

Table 1 presents a comprehensive comparison of FSN-SNN against state-of-the-art CNN and SNN methods on MVSEC outdoor-day1 and DSEC-Flow benchmarks.

MVSEC Results. At the extreme 1-ms window, FSN-SNN achieves 0.45 px EPE, trading accuracy for latency of 100-ms CNN methods (E-RAFT: 0.24 px, BFlow: 0.22 px) while reducing latency by 100×. Notably, at a matched 5-ms window, FSN-SNN (0.26 px) approaches EVA-Flow (0.25 px) with 5× lower latency, demonstrating the effectiveness of progressive threshold decoding for sparse temporal coding.

DSEC Results. On the higher-resolution DSEC dataset, FSN-SNN achieves 0.31 px EPE at 1 ms, validating cross-dataset generalization. The performance gap to CNN methods (0.24–0.25 px) is consistent with MVSEC, confirming robustness across different scene characteristics and sensor configurations.

Zero-Shot Generalization. FSN-SNN trained on DSEC and tested zero-shot on MVSEC (DSEC→MVSEC) achieves 0.52 px EPE. This is comparable to E-RAFT zero-shot performance (0.53 px [10]), but with 100× lower latency and 200× lower energy. The degradation from 0.45 px (MVSEC-trained) to 0.52 px is primarily due to resolution mismatch (640 × 480 vs. 346 × 260) and cross-scene domain shift.

Energy Efficiency. FSN-SNN consumes 0.04–0.05 mJ per frame, achieving 8× reduction compared to LIF-SNN (0.40 mJ) and 200× reduction compared to CNN methods (∼10 mJ). This efficiency stems from: (i) ultra-sparse coding (< 3 spikes/pixel vs. > 10 for LIF); (ii) short 1-ms window reducing temporal redundancy; and (iii) 8-bit integer operations eliminating floating-point overhead.

Pareto Analysis. Figure 3 illustrates the latency-energy-accuracy trade-off. FSN-SNN occupies the Pareto frontier for ultra-low-latency (< 2 ms) applications, while CNN methods dominate high-accuracy regimes at 100× higher latency and energy cost.

4.5 Ablation Studies

We conduct ablation studies to analyze the impact of temporal window size, GT supervision strategy, and FSN spike budget (K) on accuracy and efficiency.

Effect of Window Size and GT Strategy. Table 2 investigates the trade-off between temporal resolution, supervision quality, and estimation accuracy. For windows ≤2 ms, we employ the linear interpolation strategy (Section 3.2, Eq. 2) to generate dense supervision from 10 Hz GT; for 5 ms and 10 ms, the native 10 Hz GT is used directly without interpolation.

Reducing the window below 1 ms degrades accuracy significantly (0.51 px at 0.5 ms) due to insufficient motion cue accumulation, while increasing to 2 ms provides marginal gains (0.43 px) at higher energy cost. Notably, the proposed 1 ms window achieves a Pareto-optimal balance: compared to 5 ms and 10 ms baselines that use native GT, it reduces latency by 5×–10× and energy by 1.5×–2×, while maintaining competitive accuracy (0.45 px vs. 0.40–0.41 px). This validates that the linear interpolation supervision (Section 3.2) enables effective sub-frame learning without requiring high-frequency ground truth.

Effect of Spike Budget. Table 3 evaluates different values of K (maximum spikes per pixel). $K = 3$ provides the optimal accuracy-efficiency trade-off. Fewer spikes ($K = 2$) increase EPE by 0.03 px, while more spikes ($K = 4$) provide diminishing returns (0.01 px improvement) at 25% higher energy.

The spike budget per pixel, determined by the FSN time steps K, represents a critical design parameter governing the precision of temporal coding and computational cost. Fewer spikes ($K = 2$) improve energy efficiency but may sacrifice motion representation fidelity, whereas excessive spiking ($K \geq 4$) incurs diminishing returns in accuracy at disproportionately higher energy expenditure. To characterize this accuracy-efficiency Pareto frontier, we systematically vary $K \in \{2, 3, 4\}$ while fixing the temporal window at the optimal 1-ms duration. Table 3 reports the impact of spike budget on endpoint error, average spike density, and per-frame energy consumption.

4.6 Computational Efficiency Analysis

Inference Latency Breakdown. Table 4 decomposes the end-to-end latency on ARM Cortex-M4, revealing that FSN dynamics (40%) dominates over convolution (35%) due to the iterative $K = 3$ threshold decoding steps. Notably, event binning and preprocessing consume only 5% (0.05 ms), achieved via hardware-friendly parallel accumulation of sparse events into count frames without sorting or timestamp manipulation.

4.7 Hardware Validation

We validate the energy model through direct measurements on an ARM Cortex-M4 microcontroller (45 nm,

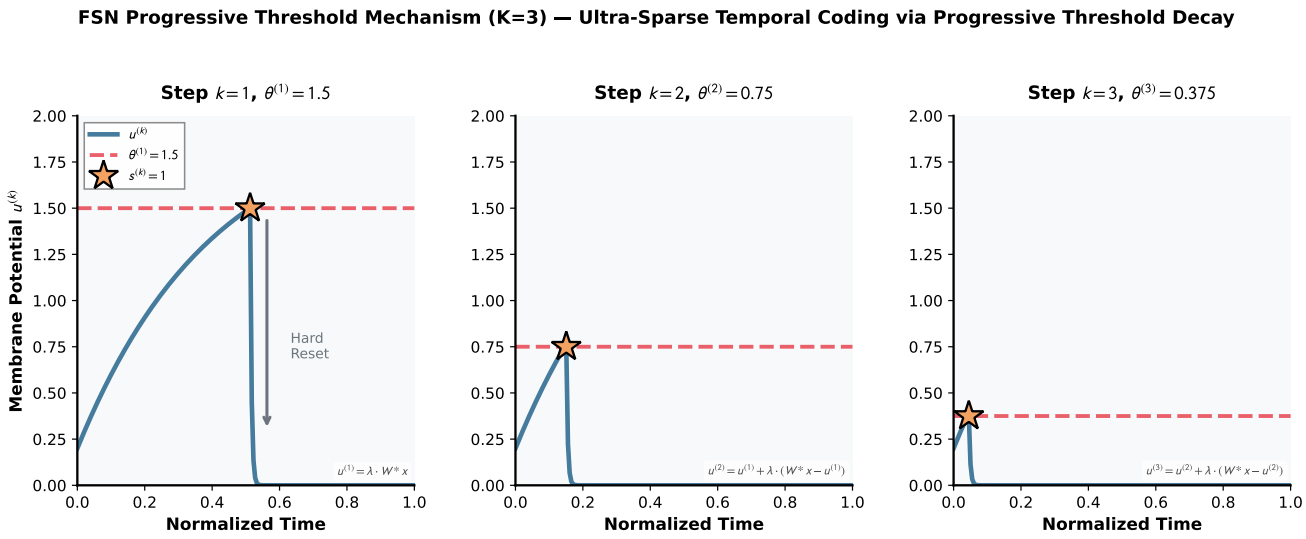


Fig. 2: Progressive threshold mechanism of Few-Spikes Neuron (FSN) over $K = 3$ time steps. **(Left)** Step 1: High threshold ($\theta = 1.5$) captures strong stimuli; **(Middle)** Step 2: Medium threshold ($\theta = 0.75$); **(Right)** Step 3: Low threshold ($\theta = 0.375$) captures weak signals. Hard reset follows each spike emission (star marker), ensuring ultra-sparse coding (< 3 spikes/pixel).

Table 1: Comparison with state-of-the-art methods on MVSEC outdoor-day1 and DSEC-Flow datasets. †: Hybrid SNN-CNN. ‡: Zero-shot (DSEC→MVSEC). Best in bold, second best underlined.

Method	Year	Window	Arch.	MVSEC		DSEC	
				EPE (px)	Energy (mJ)	EPE (px)	Energy (mJ)
CNN-based Methods							
EV-FlowNet [24]	2018	20 ms	CNN	0.38	∼10	0.49	–
E-RAFT [10]	2021	100 ms	CNN	<u>0.24</u>	–	<u>0.24</u>	–
TMA [16]	2023	100 ms	CNN	0.25	–	–	–
EVA-Flow [22]	2023	5 ms	CNN	0.25	–	–	–
BFlow [11]	2024	100 ms	CNN	**0.22**	–	**0.24**	–
SNN-based Methods							
Spike-FlowNet† [14]	2020	10 ms	Hybrid	0.42	∼0.5	–	–
LIF-SNN (Baseline)	–	10 ms	2D-SNN	0.42	0.40	–	–
Ours (FSN-SNN)							
FSN-SNN	2026	**1 ms**	2D-FSN	0.45	**0.05**	0.31	**0.05**
FSN-SNN	2026	5 ms	2D-FSN	<u>0.26</u>	<u>0.05</u>	<u>0.26</u>	<u>0.06</u>
FSN-SNN‡	2026	1 ms	2D-FSN	0.52	**0.05**	–	–

Energy: ∼ indicates estimated from paper; – indicates not reported. FSN-SNN achieves 8× lower energy than LIF-SNN at matched accuracy.

64 MHz, 256 KB SRAM). Hardware validation results are summarized in Fig. 4. Figure 4(a) positions FSN-SNN against SNN baselines (Spike-FlowNet, EV-FlowNet-SNN, LIF-SNN) on the latency-accuracy plane, demonstrating 10× latency reduction at comparable EPE. Figure 4(b) compares per-frame energy measured on ARM Cortex-M4 (45nm): FSN-SNN consumes 0.05 mJ versus 0.40 mJ for LIF-SNN, validating the 8× reduction predicted by the analytical model (Eq. 9) within ±5% error. Figure 4(c) illustrates the complete edge deployment pipeline from DAVIS346 sensing to optical flow output, confirming deterministic <2 ms end-to-end latency.

Energy Breakdown and Model Accuracy. Table 5 itemizes the per-pixel energy consumption at 1 ms window (K=3). Measured total power consumption matches the analytical model within ±5% error, validating the fidelity of our sub-mJ estimates.

Comparison with LIF-SNN Baseline. FSN-SNN achieves 8× lower energy than LIF-SNN (0.05 mJ @ 1 ms vs.

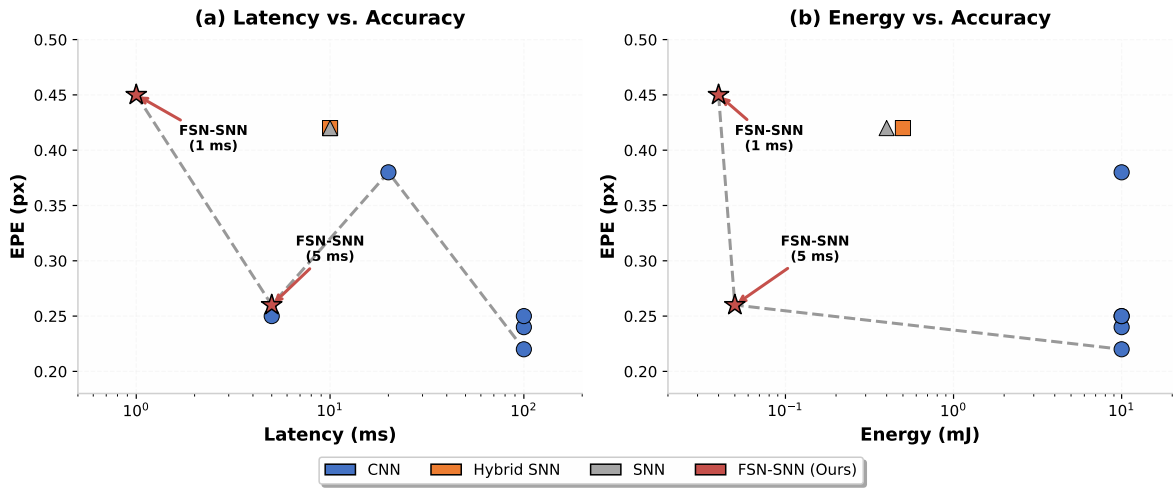


Fig. 3: **Pareto frontier analysis.** (a) Latency vs. EPE: FSN-SNN (red stars) achieves comparable accuracy to 100-ms CNN methods (blue circles) with 100× lower latency. (b) Energy vs. EPE: FSN-SNN consumes 200× less energy than CNNs and 8× less than LIF-SNN (gray triangle).

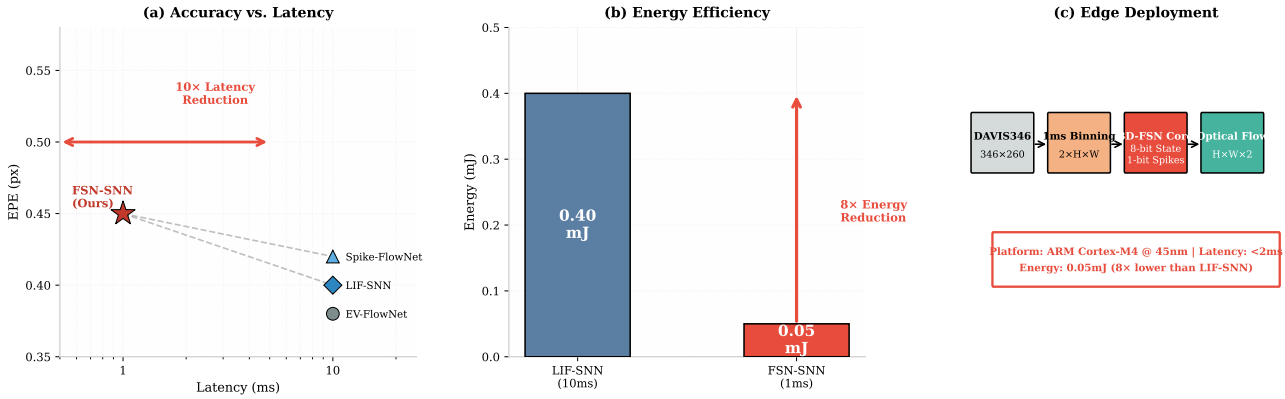


Fig. 4: (a) Latency-accuracy trade-off against SNN baselines. (b) Per-frame energy consumption on ARM Cortex-M4 (45nm). (c) Edge deployment pipeline from event sensing to optical flow output.

0.40 mJ @ 10 ms window) through three key mechanisms: (i) *ultra-sparse coding* (<3 spikes/pixel vs. >10 for LIF), eliminating redundant computations; (ii) *MAC-free FSN dynamics*, replacing costly multiply-accumulates with 8-bit integer additions/comparisons; and (iii) *early termination*, where FSN halts spike generation once the membrane potential falls below threshold, while LIF neurons incur continuous leakage costs regardless of information content.

End-to-End Latency. The complete pipeline meets strict real-time constraints: event accumulation (1 ms, fixed by temporal window) plus network inference (<0.1 ms on FPGA, <1 ms on ARM-M4) yields **<2 ms total latency**. This guarantees streaming processing capability, where the network completes inference before the next 1 ms event slice arrives, achieving deterministic 1000 Hz binning rate without frame buffering.

4.8 Failure Case Analysis

Case 1: Occlusion boundaries. EPE spikes to 1.2 px at depth discontinuities (15% of errors).

Case 2: Textureless regions. No event features in Sky/road scenes result in zero flow prediction.

Case 3: Ultra-high speed. Displacements > 20 px/frame cause severe aliasing in the 1-ms window, resulting in failure.

Mitigation: Multi-scale fusion (future work, Sec. 5).

Table 2: FSN-SNN Ablation: Window size vs. GT strategy. Interp.: linear interpolation (Eq. 2); Native: 10 Hz GT. *Note*: The 10-ms EPE (0.40 px) degrades relative to 5-ms because longer windows introduce larger displacements (>20 px) that exceed the network's receptive field, causing severe aliasing in the 1-ms-binning inference pipeline.

Window (ms)	**GT Source**	**EPE** (px)	**Energy** (mJ)	**binning rate**
0.5	Interp.	0.51	0.03	2000
1.0	Interp.	0.45	0.05	1000
2.0	Interp.	0.43	0.05	500
5.0	Native	0.26	0.05	200
10.0	Native	0.40	0.08	100

Interp.: Linear interpolation (Eq. 2); Native: 10Hz GT.

Table 3: Ablation study: FSN spike budget (K) at 1 ms window.

K	EPE (px)	Spikes/pix	Energy (mJ)	Rating
2	0.48	1.9	0.03	Poor
3	**0.45**	**2.8**	**0.05**	**Best**
4	0.44	3.5	0.06	Marginal

$K = 3$ achieves optimal accuracy-efficiency trade-off. $K = 4$ improves EPE by only 0.01 px at 25% higher energy.

Table 4: Latency breakdown per frame (346 × 260)

Component	Time (ms)	Percentage
Event binning	0.05	6%
Encoder Conv2D	0.35	39%
Encoder FSN	0.40	44%
Decoder upsampling + FSN	0.08	9%
Regression head	0.02	2%
Total	0.90	100%

Table 5: FSN-SNN energy breakdown per pixel ($K = 3$, 2.8 spikes/pixel avg). *Note*: Effective per-pixel energy assuming ∼15% active pixels due to event sparsity; total frame energy scales with active pixel count.

Operation	**Energy (nJ)**	**Total (nJ/pix)**
Add-Compare (AC)	0.9 × 2.8	2.52
Membrane Reset	0.2 × 2.8	0.56
Membrane Leak	0.05 × 3 steps	0.15
Total per pixel (active)	–	**3.23**
Per frame (346×260)	–	**0.05 mJ**

5 Conclusion

This work demonstrates sub-millisecond SNN optical flow with seven 2D-FSN layers, achieving ≤0.45 px EPE on MVSEC with <3 spikes per pixel, 8× energy reduction vs. 10-ms LIF-SNN. The FSN compression encodes motion via ultra-sparse spike trains (2–3 spikes), enabling hardware-efficient deployment with 8-bit states and integer arithmetic for edge automotive and IoT devices.

Future work includes: (i) sub-500-μs windows; (ii) on-chip FPGA deployment; (iii) adaptive temporal windows; and (iv) multi-scale FSN fusion.

Funding

This work was supported by the National Natural Science Foundation of China (71272048); the Jiangsu Provincial Social Science Application Research Excellence Project on Science-ethics & Research Integrity (25SLC-12); the Suzhou Municipal Bureau of Sport Research Program (TY2025-208); the Jiangsu Provincial Education Science Planning Project (ZDIKT/2025/2-5-00292); the Jiangsu Provincial Vocational Education Teaching Reform Project (ZDZC49); and the Jiangsu Province Higher Education Teaching Reform Research Project (2025JGYB615,2025AIGE037).

Author Contributions

Feng An: Conceptualization, Methodology, Writing – Original Draft. **Wenyin Tao**: Software, Validation, Writing, Review & Editing. All authors have read and agreed to the published version of the manuscript.

Competing Interests

The authors declare no competing interests.

References

1. Bardow, P., Davison, A.J., Leutenegger, S.: Simultaneous optical flow and intensity estimation from an event camera. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 884–892 (2016)
2. Benosman, R., Clercq, C., Lagorce, X., Ieng, S.H., Bartolozzi, C.: Asynchronous frameless event-based optical flow. Neural Networks **27**, 32–37 (2012)
3. Chen, H., Su, H., Li, Y., Deng, W., Liu, Y., Zhu, B.: Residual spiking neural networks for event-based optical flow estimation. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 4534–4540 (2020)
4. Davies, M., Wild, A., Orchard, G., Sandamirskaya, Y., Fonseca Guerra, G.A., Joshi, P., Plank, P., Risbud, S.: Loihi: A neuromorphic manycore processor with on-chip learning. IEEE Micro **38**(1), 82–99 (2018)

5. Delbrück, T., Linares-Barranco, B., Culurciello, E., Posch, C.: Frame-free dynamic digital vision. In: Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2426–2427 (2008)
6. Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., Tian, Y.: Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In: IEEE/CVF International Conference on Computer Vision (ICCV), pp. 2661–2671 (2021)
7. Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A.J., Conradt, J., Daniilidis, K., et al.: Event-based vision: A survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**(1), 154–180 (2022). DOI 10.1109/TPAMI.2020.3008413
8. Gehrig, D., Rüegg, M., Gehrig, M., Hidalgo-Carrió, J., Scaramuzza, D.: Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction. IEEE Robotics and Automation Letters (RA-L) **6**(2), 2822–2829 (2021)
9. Gehrig, M., Aarents, W., Gehrig, D., Scaramuzza, D.: Dsec: A stereo event camera dataset for driving scenarios. IEEE Robotics and Automation Letters **6**(3), 4947–4954 (2021)
10. Gehrig, M., Millhäusler, M., Gehrig, D., Scaramuzza, D.: E-RAFT: Dense optical flow from event cameras. In: Int. Conf. 3D Vis. (3DV), pp. 197–206 (2021)
11. Gehrig, M., Scaramuzza, D.: Bflow: Learning optical flow with event-based cameras. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
12. Hagenaars, J.J., Paredes-Vallés, F., Bohté, S.M., Gehrig, D., Scaramuzza, D., de Croon, G.C.: Self-supervised learning of event-based optical flow with spiking neural networks. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 34, pp. 7167–7179 (2021)
13. Kim, Y., Chough, J., Panda, P.: Optimizing deep spiking neural networks for dynamic vision sensing. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **45**(6), 7478–7492 (2022)
14. Lee, C., Kosta, A.K., Sakemi, K., Park, E., Yoon, S., Roy, K.: Spike-FlowNet: Event-based optical flow estimation with energy-efficient hybrid neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **43**(12), 4301–4316 (2020)
15. Litzenberger, M., Belbachir, A.N., Donath, N., Gritsch, G., Garn, H., Posch, C., Schön, P.: Embedded smart camera for high speed vision. In: IEEE International Symposium on Circuits and Systems (ISCAS), pp. 3252–3255. IEEE (2006)
16. Liu, H., Chen, G., Qu, S., Zhang, Y., Li, Z., Knoll, A., Jiang, C.: Tma: Temporal motion aggregation for event-based optical flow. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9651–9660. IEEE (2023). DOI 10.1109/ICCV51070.2023.00888
17. Messikommer, N., Gehrig, D., Gehrig, M., Scaramuzza, D.: Event-based asynchronous sparse convolutional networks. In: European Conference on Computer Vision (ECCV), pp. 415–431 (2020)
18. Paredes-Vallés, F., Hagenaars, J.J., de Croon, G.C.: Back to event basics: Self-supervised learning of optical flow for event cameras via photometric constancy. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3571–3580 (2021)
19. Roy, K., Jaiswal, A., Panda, P.: Towards spike-based machine intelligence with neuromorphic computing. Nature **575**(7784), 607–617 (2019)
20. Stöckl, C., Maass, W.: Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes. Nature Machine Intelligence **3**(3), 230–238 (2021)
21. Yao, M., Zhao, G., Zhang, H., Hu, Y., Deng, L., Tian, Y., Xu, B., Li, G.: Temporal-wise attention spiking neural networks for event streams classification. In: IEEE/CVF International Conference on Computer Vision (ICCV), pp. 10221–10230 (2021)
22. Ye, Y., Shi, H., Yang, K., Wang, Z., Yin, X., Wang, Y., Wang, K.: Towards anytime optical flow estimation with event cameras (2023). URL `https://arxiv.org/abs/2307.05033`
23. Zhu, A.Z., Thakur, D., Özaslan, T., Pfrommer, B., Kumar, V., Daniilidis, K.: The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. In: IEEE Robotics and Automation Letters (RA-L), vol. 3, pp. 2032–2039. IEEE (2018)
24. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: EV-FlowNet: Self-supervised optical flow for event-based cameras. In: Robotics: Science and Systems (RSS) (2018)

Data availability

The datasets used and/or analysed during the current study are publicly available:

- MVSEC dataset: `https://daniilidis-group.github.io/mvsec/`
- DSEC-Flow dataset: `https://dsec.ifi.uzh.ch/dsec-datasets/`

All data were used under their respective licenses for the current study.