

FracEvent: Event-Camera Simulation via Fractional-Relaxation Pixel Dynamics

Langyi Chen¹ Chuanzhi Xu^{1*} Haoxian Zhou¹ Pengfei Ye² Ziyu Luo³
 Haodong Chen¹ Qiang Qu¹ Xiaoming Chen³ Weidong Cai¹
¹The University of Sydney ²Massachusetts Institute of Technology
³Beijing Technology and Business University

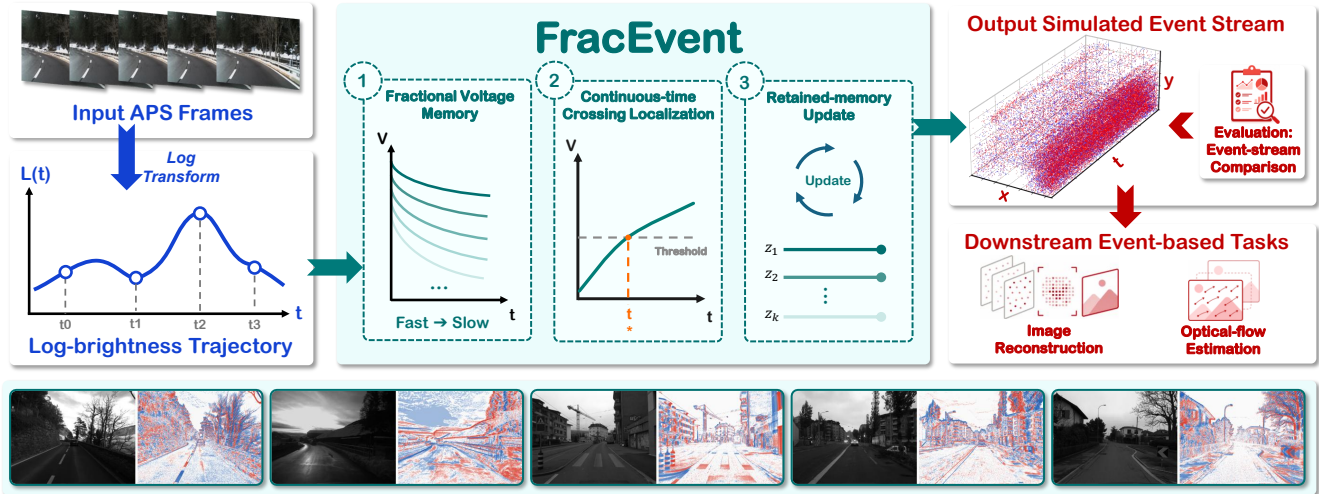


Figure 1. **Overview.** Given input APS frames, FracEvent converts the resulting log-brightness trajectory into an event stream through three sensor-side mechanisms: fractional voltage memory, continuous-time crossing localization, and retained-memory updates. The generated event stream is evaluated through event-stream comparison and downstream event-based tasks.

Abstract

Event cameras asynchronously report brightness changes with microsecond-level temporal resolution, but real event data remain difficult to collect at scale because specialized sensors, careful synchronization, and task-specific annotations are required. Event-camera simulation is therefore important to event-based vision tasks. Most practical simulators build on contrast-threshold event generation, some with additional filtering, stochastic noise, or hand-tuned sensor parameters. While effective, such formulations often simplify the temporal structure produced by the lifecycle of each pixel, which can distort event timing and weaken downstream transfer. We introduce FracEvent, an event simulator that models this pixel-level lifecycle with fractional-relaxation voltage dynamics. Given a log-intensity trajectory, FracEvent drives a compact stack of relaxation modes, combines their responses into a voltage state, emits ON/OFF events by localizing threshold crossings on the continuous voltage trajectory, and updates the

reference while retaining the underlying memory modes. This retained state links residual voltage response to later event timing. We evaluate FracEvent through event-stream comparison and downstream transfer on image reconstruction and optical flow estimation. Across multiple datasets, FracEvent improves the temporal structure of generated events and achieves stronger downstream-transfer results than competing simulator baselines, showing its practical value for event-camera simulation. [Project page](#)

1. Introduction

Event cameras, also known as dynamic vision sensors (DVS), report brightness changes as asynchronous pixel events. At each pixel, the sensor emits a positive (On) or negative (OFF) event when the accumulated log-brightness change crosses a contrast threshold. This threshold-triggered readout provides microsecond-level temporal resolution and high dynamic range while concentrating output around temporal brightness changes [2, 18]. These properties make event sensing valuable for robotics, au-

*Corresponding author: chuanzhi.xu@sydney.edu.au

onomous driving, optical flow estimation, and video reconstruction [6, 36]. Representative systems use events for visual-inertial odometry, steering prediction, joint optical-flow/intensity estimation, and event-to-image reconstruction [1, 21, 28, 41]. However, collecting real event streams at scale requires dedicated event sensors and capture hardware, often with task-specific synchronization, annotation, and scene setups [8, 42]. Because event cameras remain relatively expensive and scarce, real event collection is hard to expand on demand across new scenes, motion conditions, and sensor assumptions [45]. At the same time, general computer vision research already provides large collections of high-quality frame-based video data. Therefore, event-camera simulation serves as an important complement to real event data by turning frame-based data into event streams [7]. However, this conversion is not easy as a simple change of data format. Frame sequences and renderers provide per-pixel brightness trajectories, while the resulting events depend on when those trajectories drive pixel states across ON/OFF thresholds. Their temporal sampling, interpolation, smoothing, and local contrast evolution can shift threshold-crossing times and change event counts [12, 15, 27].

This trajectory-conditioned view separates event simulation into two coupled components. The first is the *input trajectory*: a renderer, high-rate video, sparse frame sequence, interpolation module, or raw-camera pipeline provides a brightness signal over time. The second is the *sensor-side conversion*: given that trajectory, each pixel must decide when its internal state emits an event. Existing methods combine these choices in different ways, from continuous rendering [25, 27] or interpolation networks applied to input frames [7], to sensor-aware models of nonidealities and noise [12, 15, 19], to learned and controllable generators [10, 13, 44]. For the sensor-side conversion, an important observation is that a real event stream is not just a set of independent threshold crossings. A pixel accumulates response, fires, updates its reference, and then continues from a state shaped by the recent past. Strong contrast may produce repeated events at the same pixel, ON and OFF thresholds may not be balanced, and finite analogue response can delay or extend activity.

In this paper, we propose FracEvent, an event-camera simulator that focuses on sensor-side conversion by modeling the pixel-level event lifecycle conditioned on a supplied log-intensity trajectory. The input log-intensity $L(t)$ drives several relaxation modes: fast modes follow immediate brightness changes, slow modes retain residual response, and their weighted sum forms a voltage state. Events are simulated from ON/OFF threshold crossings of this voltage trajectory relative to a reference. After the event, the reference moves, while the memory modes remain active. Figure 1 summarizes this flow from input trajectory to sim-

ulated events and evaluation.

This design targets three structures visible in real event data: repeated same-pixel triggering, threshold-crossing times within frame intervals, and ON/OFF asymmetry. We evaluate FracEvent through event-stream comparison and downstream transfer on image reconstruction and optical flow estimation. For event-stream comparison, we compare simulated event streams against real event-camera data using quantitative event statistics, qualitative visualizations, and matched-window component ablations. We then test downstream transfer in two settings: image reconstruction with an E2VID-style model and established reconstruction evaluation practice [4, 28, 30, 32], and optical flow estimation with an EV-FlowNet-style training and evaluation protocol [42, 43]. Across these evaluations, FracEvent gives stronger downstream transfer than competing synthetic event sources, with the best reconstruction MSE and LPIPS and the lowest optical-flow mean AEE among models trained on simulated events.

Our contributions can be summarized as follows:

- We introduce FracEvent, an event-camera simulator that models the **pixel event lifecycle** with **fractional-relaxation voltage memory**, preserving residual sensor state across repeated ON/OFF events.
- We formulate **trajectory-conditioned sensor-side conversion** for event simulation, allowing the same pixel dynamics to be paired with renderers, interpolation modules, frame sequences, or other sources that supply log-intensity trajectories.
- We develop a **continuous-time event-generation module** with closed-form mode updates, active-pixel bisection for triggered threshold crossings, event-wise reference updates, and separate ON/OFF thresholds.
- We validate FracEvent through direct **event-stream comparison**, component ablations, and **downstream transfer**, obtaining the lowest IEI distance and polarity error, the best reconstruction MSE and LPIPS, and the lowest mean AEE among models trained on simulated events for optical flow estimation under fixed protocols.

2. Related Work

Input Trajectories for Event Simulation. Event simulation begins with the brightness trajectory that drives each pixel. Rendering-based simulators such as ESIM obtain this signal from continuous scene motion before applying an event rule, and later systems extend the same route to robotics, path tracing, contact-rich simulation, and task-specific synthetic data [11, 14, 25, 27, 29]. Video-to-event pipelines take a different route by reusing frame datasets. In these pipelines, interpolation, exposure handling, filtering, and raw-camera processing shape the upstream trajectory [7, 12, 15, 24, 39]. The quality of this trajectory matters because missing motion, occlusion, or exposure changes can-

not be recovered by any later thresholding rule. FracEvent addresses the complementary problem: given $L(t)$ from a renderer, high-rate video, interpolation module, or raw-camera pipeline, it isolates the sensor-side conversion from the supplied trajectory to events.

Sensor-Side Event Simulation. Given a brightness trajectory, the remaining question is how each pixel converts it into asynchronous ON/OFF events. Classical DVS and DAVIS models emit events when accumulated log-brightness change crosses a contrast threshold [2, 18], while practical simulators add interpolation, bandwidth limits, threshold mismatch, noise, and calibration [12, 15, 27]. Circuit-aware and stochastic simulators model event generation through voltage or first-passage processes, emphasizing bias, threshold variability, and sensor nonidealities [19]. Learned and domain-adaptive generators instead match event distributions or synthesize controllable event streams from data-driven priors [10, 13, 44]. FracEvent follows the explicit sensor-side modeling direction, using deterministic fractional-relaxation voltage memory, continuous-time threshold crossing, retained post-event mode state, and separate ON/OFF thresholds, while keeping the mechanism inspectable in equations and ablations.

Downstream Applications of Simulated Event Streams. Simulated events also support downstream event-based vision tasks, whose representations and temporal structure are central to 3D reconstruction, event-stream super-resolution, and pose estimation [35, 37, 40]. Here we focus on image reconstruction and optical flow because they provide mature and complementary evaluations: reconstruction tests whether accumulated events preserve recoverable appearance, while flow tests whether event timing and local motion structure support motion estimation. For image reconstruction, E2VID- and FireNet-style recurrent models map event tensors to intensity images [28, 30], with later work systematizing reconstruction protocols and sim-to-real reporting [4, 5, 32]. For optical flow, EV-FlowNet learns dense flow from events with self-supervised photometric losses and evaluates on MVSEC [42, 43]; later work adds stronger architectures, higher-resolution driving datasets, and contrast-maximization objectives [8, 9, 16]. We therefore evaluate both tasks with the network, representation, schedule, and test protocol fixed while only the simulated event source changes.

3. Methodology

3.1. Problem Setup for Event-stream Simulation

At each pixel, let $I(t)$ denote a normalized intensity trajectory supplied by an upstream source, such as a frame sequence, interpolation module, or renderer. We use the log-intensity form common in contrast-threshold event-camera

models:

$$L(t) = \log(I(t) + i_{\text{dark}} + \epsilon). \quad (1)$$

Here i_{dark} and ϵ are small positive offsets that stabilize the log transform near zero intensity. Given $L(t)$, FracEvent generates an asynchronous event stream:

$$\mathcal{E} = \{(x_i, y_i, t_i, p_i)\}_{i=1}^N, \quad (2)$$

where (x_i, y_i) is the pixel coordinate, t_i is the event timestamp, and $p_i \in \{+1, -1\}$ is the event polarity, with $p_i = +1$ denoting an ON event and $p_i = -1$ denoting an OFF event.

Fixing $L(t)$ separates the upstream trajectory from the pixel rule that converts brightness changes into events. This lets the same sensor-side model be paired with different upstream sources while isolating how pixel dynamics affect timing, polarity, and density. Real event streams often contain repeated same-pixel triggers and short bursts, so the pixel rule must carry residual state beyond the current frame-to-frame contrast. We instantiate this rule with fractional-relaxation voltage memory, continuous-time threshold crossing, and reference updates that retain the memory state. Figure 2 shows how these components form this sensor-side conversion model.

3.2. Fractional-Relaxation Pixel Dynamics

FracEvent builds voltage memory from multiple fractional-relaxation modes. The modes are indexed by log-time offsets:

$$u_m \in [-R, R], \quad m = 1, \dots, M, \quad (3)$$

sampled uniformly over the configured relaxation log range R . For a fractional shape parameter $\alpha \in (0, 1)$, the mode weights are:

$$\tilde{w}_m = \frac{\sin(\pi\alpha)}{\cosh(\alpha u_m) + \cos(\pi\alpha)}, \quad w_m = \frac{\tilde{w}_m}{\sum_j \tilde{w}_j}, \quad (4)$$

where the denominator normalizes the weights over all modes. For example, $\alpha = 1$ corresponds to the ordinary first-order single-mode case, with $M = 1$ and $w_1 = 1$.

For mode m , let $z_m(t)$ be a memory channel with time constant τ_m . Each channel follows a first-order relaxation equation:

$$\frac{dz_m(t)}{dt} = \frac{L(t) - z_m(t)}{\tau_m}. \quad (5)$$

The pixel voltage, denoted by $v(t)$, is the weighted sum of these channels:

$$v(t) = \sum_{m=1}^M w_m z_m(t). \quad (6)$$

Together, the weighted channels form a compact finite-mode approximation to a broad fractional-memory kernel. This distributed-exponential construction follows standard fractional-relaxation representations and the classical distributed-relaxation view behind Cole-Cole dispersion

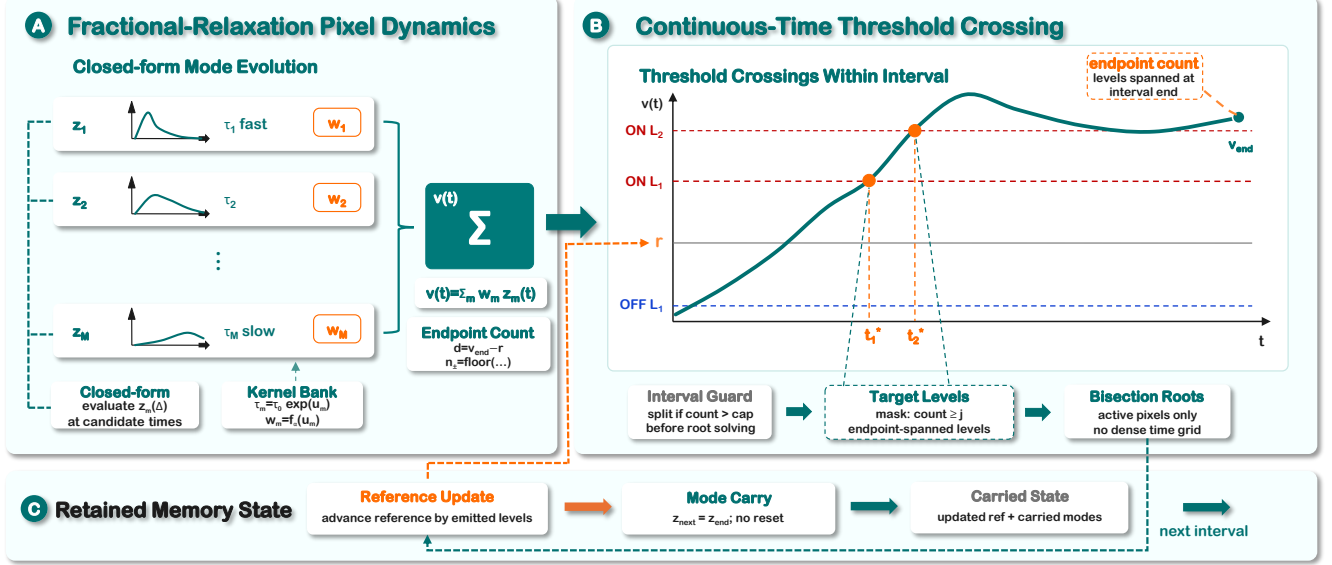


Figure 2. **Equivalent pixel-dynamics view of FracEvent.** The log-intensity trajectory drives a finite fractional-memory mode stack whose weighted sum forms the sensor-side event-generation voltage state. Events are emitted by continuous-time ON/OFF threshold crossing relative to a reference state. After an event, only the reference is updated while the memory modes are preserved.

[3, 20, 26]. Fast modes track immediate brightness changes, while slow modes preserve residual response over longer intervals. For each frame interval, the mode time constants are set from the mid-interval brightness:

$$\tau_0 = \tau_{\text{ref}} \left(\frac{I_{\text{ref}}}{\bar{I} + i_{\text{dark}}} \right)^\beta, \quad \tau_m = \tau_0 e^{u_m}, \quad (7)$$

where $\bar{I} = (I(t_k) + I(t_{k+1}))/2$, τ_{ref} and I_{ref} set the response scale and reference brightness, and β controls the brightness dependence. We bound τ_0 to $[\tau_{\text{min}}, \tau_{\text{max}}]$ to avoid unrealistically slow or fast responses. The offsets u_m then spread this base time constant across the memory modes. At sequence start, all modes are initialized from the first log-intensity frame, and the reference is initialized as their weighted sum.

After τ_m is fixed for an interval, each memory channel can be evaluated in continuous time. We assume a linear log-intensity segment within the interval:

$$L(t_k + \Delta) = L_0 + s\Delta, \quad 0 \leq \Delta \leq dt. \quad (8)$$

Here $dt = t_{k+1} - t_k$, $L_0 = L(t_k)$, $s = (L(t_{k+1}) - L_0)/dt$, and $a_m(\Delta) = e^{-\Delta/\tau_m}$. Each mode then has the closed-form update:

$$z_m(t_k + \Delta) = a_m(\Delta)z_m(t_k) + (1 - a_m(\Delta))L_0 + s[\Delta - \tau_m(1 - a_m(\Delta))]. \quad (9)$$

This expression gives the voltage state at any candidate time in the interval. The threshold solver below uses this continuous-time state to assign event timestamps without a dense temporal grid.

3.3. Continuous-Time Threshold Crossing

FracEvent emits events when the voltage state crosses threshold levels relative to the reference. Let r denote the reference voltage carried into the current interval, and define the reference-relative voltage:

$$u(t) = g(v(t) - r), \quad (10)$$

where g is a scalar gain set to one in the reported experiments. An ON event is generated when $u(t) \geq \theta_{\text{on}}$, and an OFF event is generated when $u(t) \leq -\theta_{\text{off}}$, where θ_{off} is stored as a positive magnitude.

For each frame interval, the solver first computes ON/OFF event counts from the endpoint reference-relative voltage:

$$n^+ = \left\lfloor \frac{[u(t_{k+1})]_+}{\theta_{\text{on}}} \right\rfloor, \quad n^- = \left\lfloor \frac{[-u(t_{k+1})]_+}{\theta_{\text{off}}} \right\rfloor. \quad (11)$$

Here $[q]_+ = \max(q, 0)$ denotes the positive part. These counts define the ON/OFF threshold levels relative to the interval-start reference. For each active pixel and each ON level $j = 1, \dots, n^+$, it solves $v(t) = r + j\theta_{\text{on}}/g$ by bisection on $[t_k, t_{k+1}]$. For each OFF level $j = 1, \dots, n^-$, it solves $v(t) = r - j\theta_{\text{off}}/g$ in the same way. The bisection stage assigns continuous-time timestamps to these counted levels for active pixels, and the resulting events are sorted by timestamp before output. This is a local root-localization step: on monotone pixel subintervals, each counted level has a unique crossing. If the endpoint count in an interval exceeds the solver cap, the interval is recursively split

Table 1. **Evaluation setup.** Datasets, protocols, and metrics are summarized for event-stream comparison and downstream tasks.

Evaluation	Dataset	Protocol	Metrics
Event-stream comparison	DAVIS240C windows [22] and DAVIS346 real-life captures	Same frame/timestamp windows and metrics, with simulated streams differing only in sensor-side dynamics.	Count, IEI distance, polarity, time surface
Image reconstruction	GoPro train [23] and DAVIS240C/HQF tests [22, 32]	Same resolution, E2VID-style model, loss, optimizer, schedule, and test sets.	MSE, SSIM, LPIPS
Optical flow estimation	MVSEC Day2 train and Day1/indoor tests [42]	EV-FlowNet-style network, event representation, losses, 300k schedule, and evaluation splits.	AEE, outlier percentage

before bisection, so dense bursts are handled without a uniform substep grid.

After emitting these levels, FracEvent advances only the reference state:

$$\begin{aligned} r &\leftarrow r + \theta_{\text{on}}/g && \text{for ON,} \\ r &\leftarrow r - \theta_{\text{off}}/g && \text{for OFF.} \end{aligned} \quad (12)$$

For multiple same-polarity levels in one interval, the same step is applied once per emitted level. The mode states $z_m(t)$ are not reset. Thus, an event shifts the reference voltage while leaving residual voltage memory available for the next crossing.

This lifecycle gives the simulated stream three useful properties. First, preserving the fractional modes carries sub-threshold response across events, which supports repeated triggering and inter-event-interval structure. Second, continuous-time bisection assigns timestamps to counted threshold levels inside each frame interval, producing asynchronous event times and avoiding grid-locked emissions. Third, separate ON/OFF reference updates represent polarity asymmetry with explicit threshold magnitudes. Together, these properties produce standard asynchronous event streams whose density, timing, and polarity are governed by explicit sensor-side dynamics.

Algorithm 1 gives the core sensor-side conversion loop; the corresponding implementation and additional details are provided in the supplementary code and Supplementary Material.

4. Experiments and Results

4.1. Experimental Setup

We evaluate FracEvent through event-stream comparison and downstream task utility. For event-stream comparison, we compare simulated event streams under matched timestamp windows and visualize the accumulated events qualitatively. For downstream task utility, we compare events simulated by FracEvent and by baseline simulators under a fixed evaluation protocol, using image reconstruction and optical flow estimation as the downstream tasks. Table 1 summarizes the datasets, protocols, and metrics. We use events and APS frames from the DAVIS240C dataset for event-stream comparison and reconstruction evaluation

Algorithm 1 Sensor-side event generation in FracEvent

Input: frames $\{I(t_k)\}_{k=0}^K$, timestamps, parameters

Output: sorted event stream \mathcal{E}

- 1: Convert $I(t_k)$ to log intensity $L(t_k)$ by Equation (1)
 - 2: Initialize $z_m \leftarrow L(t_0)$ for all modes and $\mathcal{E} \leftarrow \emptyset$
 - 3: Set $r \leftarrow v(t_0)$ by Equation (6)
 - 4: **for** $k = 0, \dots, K - 1$ **do**
 - 5: Set interval time constants τ_m by Equation (7)
 - 6: Define $L(t)$ on $[t_k, t_{k+1}]$ by Equation (8)
 - 7: Evaluate endpoint voltage $v(t_{k+1})$ with Equations (6) and (9)
 - 8: Store $r_0 \leftarrow r$
 - 9: Compute endpoint counts n^+, n^- using Equation (11)
 - 10: **if** $\max(n^+, n^-)$ exceeds the solver cap **then**
 - 11: Recursively process the two split subintervals
 - 12: **continue**
 - 13: **for** each active pixel and counted threshold level **do**
 - 14: Locate the event timestamp t^* by bisection
 - 15: Append (x, y, t^*, p) to \mathcal{E}
 - 16: Update $r \leftarrow r_0 + n^+ \theta_{\text{on}}/g - n^- \theta_{\text{off}}/g$ pixelwise
 - 17: Carry the mode states $z_m(t_{k+1})$ to the next interval
 - 18: Sort \mathcal{E} by timestamp and return it
-

[2, 22]. GoPro high-frame-rate video supplies input for matched simulated event generation [23], HQF serves as an additional image reconstruction test set [32], and MVSEC contains event streams, APS frames, and ground-truth flow for optical flow estimation training and evaluation [42].

For simulator baselines, we compare with the widely used simulators ESIM [27], v2e [12], and DVS-Voltmeter [19], using the calibrated or recommended public settings provided by each implementation. This keeps the baselines at their intended public operating points rather than applying our event-statistics calibration objective to competing simulators. Because the downstream tasks use datasets captured by DAVIS240-series and DAVIS346-series event cameras, FracEvent uses separate parameter settings for the two sensor families. We calibrate a single parameter set for each sensor family on a small set of DAVIS240C and MVSEC windows, matching only basic event-stream statis-

Table 2. **Event-stream results on DAVIS240C.** Simulated event streams are compared with real DAVIS240C events over matched windows. Bold values mark the best value for each metric, with count ratio closest to 1.

Method	Count ratio	IEI dist. ↓	Polarity ↓	Time surf. ↓
FracEvent	1.138	0.0964	0.0244	0.5551
DVS-Voltmeter	1.101	0.1271	0.0895	0.4791
v2e	1.366	0.3040	0.0828	0.5187
ESIM	9.221	0.6365	0.0800	0.5349

tics. The resulting settings are DAVIS240 ($\alpha = 0.76$, $M = 4$, $\tau_{\text{ref}} = 0.0025$ s, $\theta_{\text{on}} = 0.5625$, $\theta_{\text{off}} = 0.4000$) and DAVIS346 ($\alpha = 0.72$, $M = 6$, $\tau_{\text{ref}} = 0.0025$ s, $\theta_{\text{on}} = 0.4500$, $\theta_{\text{off}} = 0.5000$). These windows are disjoint from the sequences and windows used for event-stream comparison. The calibration does not use downstream image reconstruction or optical flow estimation losses, downstream test annotations, image-reconstruction metrics computed against target frames, or any downstream evaluation metric, and the selected parameter settings are fixed before simulating the training event streams. More details on the experimental settings are provided in Supplementary Material Secs. A-C.

4.2. Event-Stream Comparison

We first compare simulated event streams with real DAVIS240C events, reporting means over 32 sampled 50 ms windows from four sequences. For each matched simulated/real pair (S, R), we use four window-level metrics:

- **Count ratio** measures event density as $(|S|+1)/(|R|+1)$, where a value of 1 indicates matched event counts.
- **IEI distance** measures same-pixel timing structure beyond total event count. Following event-interval comparisons in prior simulator evaluations [19], we sort events at each pixel, collect consecutive intervals $\Delta t > 0$, and compare the simulated and real event distributions of

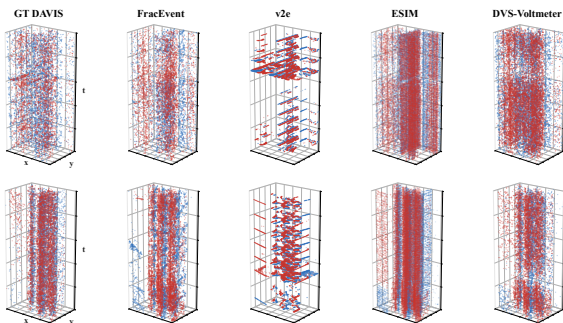


Figure 3. **Qualitative event-stream comparison.** Each row shows a spatiotemporal event plot for one 50 ms DAVIS240C window, with columns comparing real events and generated events from the same interval. Red and blue denote ON and OFF events.

$\log_{10} \Delta t$ with the 1-Wasserstein distance [33].

- **Polarity** measures ON/OFF balance as $|\Pr_S(p = +1) - \Pr_R(p = +1)|$.
- **Time surface** measures whether recent events occur at similar pixels and polarities. For each polarity p , we build the standard exponential time surface $T_p(x, y) = \exp(-(t_1 - t_p(x, y))/\tau)$, where $t_p(x, y)$ is the latest event time at pixel (x, y) [17, 31]. We compute the correlation ρ between simulated and real event time surfaces and report $1 - \max(\rho, 0)$.

Baseline Comparison. Table 2 compares simulated event streams with real DAVIS240C events over multiple windows. FracEvent gives the lowest IEI distance and the lowest polarity error, while DVS-Voltmeter gives the closest event count and the lowest time-surface distance. The visualization in Figure 3 shows the same trend. FracEvent keeps events continuously distributed over time and close to the real DAVIS240C pattern. In contrast, v2e and ESIM concentrate events around discrete timestamps, and ESIM also produces overly dense events. Time-surface distance complements these metrics by focusing on where recent events occur. Thus, the best time-surface score does not necessarily imply a lower IEI distance or stronger repeated-trigger timing.

Ablation Study. Under the same window protocol, each ablation removes one component of FracEvent while keeping the input frames, parameter settings, and metrics fixed.

Table 3 reports this component comparison. Full FracEvent gives the closest event-count match, the lowest IEI distance, and the lowest polarity error. Removing multi-mode memory or retained memory state mainly worsens the IEI distance, while removing separate thresholds leaves the timing metric nearly unchanged but sharply worsens polarity balance.

We also capture additional real-life scenes with a DAVIS346 event camera. Figure 4 compares APS frames, real events, and simulated streams under the same win-

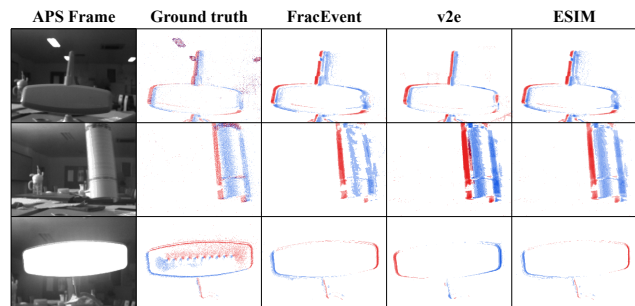


Figure 4. **DAVIS346 real-camera comparison in real-life scenes.** Each scene shows the APS frame, the real events, and the events simulated by FracEvent, v2e, and ESIM for the corresponding frame interval.

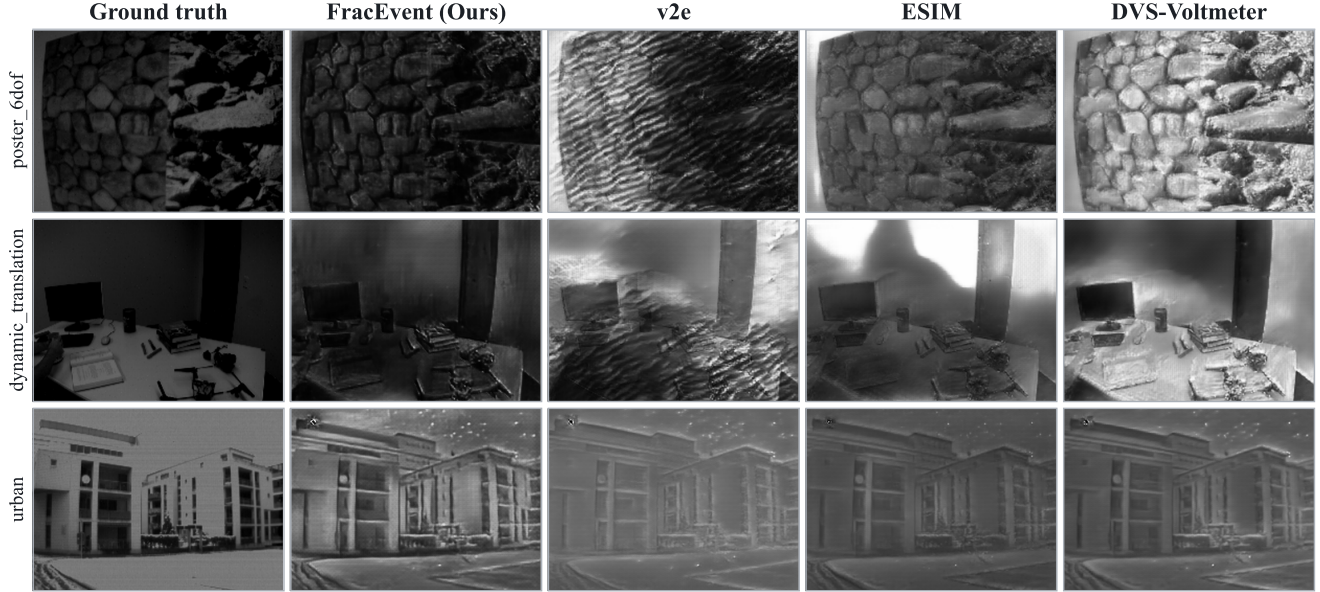


Figure 5. **Qualitative image reconstruction comparison on DAVIS240C.** The figure shows three DAVIS240C examples and compares image reconstruction results from models trained with event streams synthesized by different baselines.

Table 3. **Ablation study on FracEvent components.** All variants are evaluated under the same matched-window protocol and event-stream metrics. Bold values mark the best value for each metric, with count ratio closest to 1.

Variant	Count ratio	IEI dist. ↓	Polarity ↓	Time surf. ↓
Full FracEvent	1.138	0.0964	0.0244	0.5551
w/o multi-mode memory	1.322	0.1369	0.0247	0.5375
w/o retained memory state	1.176	0.1244	0.0258	0.5559
w/o separate thresholds	1.179	0.0976	0.0978	0.5551

dows. FracEvent better preserves the real event contours and ON/OFF polarity structure, while v2e misses parts of the event pattern and ESIM tends to produce denser edge responses. More details are provided in Supplementary Material Sec. D.

4.3. Image Reconstruction

We first test downstream transfer on image reconstruction. For each simulated event source, we train the same E2VID-style recurrent reconstruction model on events generated from the full GoPro training split at 240 fps, resized to 240×180 [23, 28]. The simulated event source is the only changing input, while the training frames, model, resolution, and optimizer budget are fixed. Each checkpoint is evaluated on full DAVIS240C sequences as the primary test set and on HQF as an additional test set [32]. We report MSE, SSIM [34], and LPIPS [38] over full sequences, following reporting style in EVREAL [4].

As shown in Table 4, on the DAVIS240C test set,

Table 4. **Image reconstruction results on DAVIS240C and HQF.** Each model is trained from one simulated event source and evaluated on DAVIS240C and HQF sequences. Bold values mark the best result in each column.

Train source	DAVIS240C			HQF		
	MSE ↓	SSIM ↑	LPIPS ↓	MSE ↓	SSIM ↑	LPIPS ↓
FracEvent	0.030	0.371	0.460	0.052	0.355	0.443
v2e	0.065	0.253	0.575	0.073	0.304	0.556
ESIM	0.054	0.411	0.496	0.080	0.372	0.551
DVS-Voltmeter	0.129	0.275	0.561	0.086	0.351	0.508

FracEvent gives the lowest MSE (0.030) and lowest LPIPS (0.460) among compared simulated training sources. The structural metric differs: ESIM has the highest DAVIS240C SSIM. HQF shows similar metric-specific behavior: FracEvent is best by MSE and LPIPS, while ESIM has the highest SSIM. Figure 5 provides representative DAVIS240C reconstruction examples under the same fixed-training comparison. More details are provided in Supplementary Material Sec. E.

4.4. Optical Flow Estimation

We then test downstream transfer on optical flow estimation. For each event source, we train the same EV-FlowNet-style self-supervised model on MVSEC outdoor_day2 for 300k steps [43]. Runs trained on simulated events use the same training sequence, while the model trained on Real MVSEC events serves as the real-event control. The model, loss, schedule, and evaluation code are fixed, and

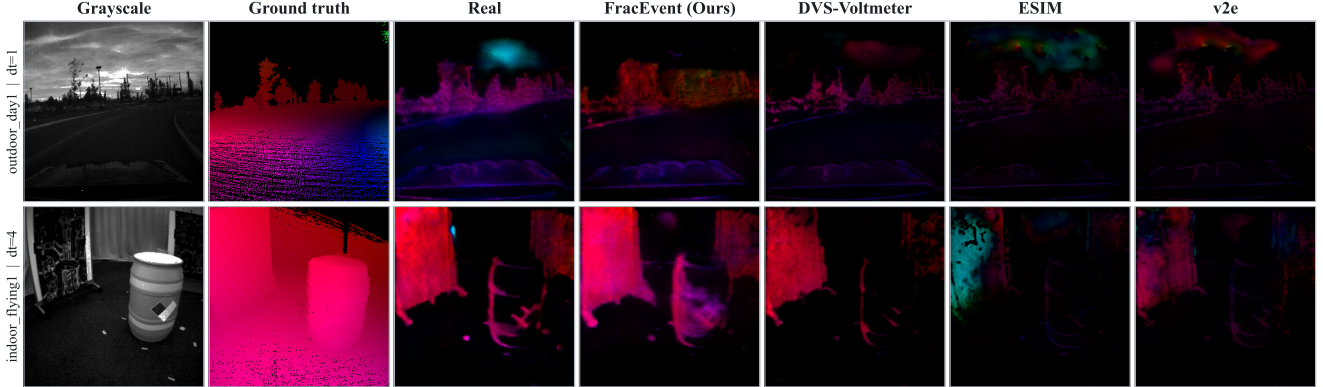


Figure 6. **Optical-flow predictions on MVSEC.** The figure shows representative outdoor and indoor evaluations. Columns compare ground truth with predictions from models trained on real events or event streams synthesized by different baselines.

Table 5. **Optical flow estimation results on MVSEC.** Each model is trained from one event source and evaluated on outdoor and indoor MVSEC sequences. Real MVSEC denotes the real-event control, and bold values mark the best AEE among simulated-event training runs in each column.

Method	Outdoor1		Indoor1		Indoor2		Indoor3		Mean	
	AEE	Out.	AEE	Out.	AEE	Out.	AEE	Out.	AEE	Out.
<i>dt = 1 frame</i>										
Real MVSEC	0.55	0.4	1.03	2.2	1.60	13.4	1.48	11.3	1.17	6.8
FracEvent	0.63	0.3	1.07	2.2	1.71	14.8	1.57	13.7	1.24	7.8
DVS-Voltmeter	0.59	0.3	1.12	2.1	1.83	16.0	1.66	12.4	1.30	7.7
ESIM	0.65	0.4	1.26	4.6	2.13	22.7	1.90	17.3	1.49	11.2
v2e	0.76	0.5	1.27	4.7	2.14	22.7	1.93	18.0	1.53	11.5
<i>dt = 4 frames</i>										
Real MVSEC	1.53	12.0	3.14	37.1	5.49	53.5	4.52	47.8	3.67	37.6
FracEvent	1.91	15.3	3.39	41.1	5.85	53.0	5.34	50.4	4.12	40.0
DVS-Voltmeter	1.85	17.7	3.87	48.7	6.44	65.4	5.62	61.2	4.44	48.3
ESIM	2.19	22.8	4.76	64.6	7.50	74.9	6.67	73.5	5.28	58.9
v2e	2.53	29.2	4.62	61.1	7.31	74.4	6.54	73.5	5.25	59.6

only the training event source changes. We evaluate on `outdoor_day1` and `indoor_flying1/2/3` at $dt = 1$ and $dt = 4$. We report average endpoint error (AEE) and event-masked KITTI-style outlier percentage, where outliers have endpoint error above 3 px and above 5% of the ground-truth flow magnitude.

As shown in Table 5, FracEvent gives the lowest mean AEE among models trained on simulated events and remains closest to the real-data control across the outdoor and indoor evaluations. Averaged over the four MVSEC evaluation sequences at both $dt = 1$ and $dt = 4$, it reaches 2.68 mean AEE, compared with 2.42 for Real MVSEC, 2.87 for DVS-Voltmeter, 3.38 for ESIM, and 3.39 for v2e. FracEvent also gives the best AEE among models trained on simulated events in six of the eight sequence AEE columns, with DVS-Voltmeter strongest on the two `outdoor_day1` columns. These results indicate that FracEvent provides the

most consistent simulated training source for EV-FlowNet-style optical flow estimation under this protocol. Figure 6 gives representative outdoor and indoor predictions. More details are provided in Supplementary Material Sec. F.

5. Discussion and Conclusion

In this paper, we presented FracEvent, a sensor-side event-camera simulator that converts a supplied log-intensity trajectory into events by modeling the pixel lifecycle: fractional voltage memory, continuous-time ON/OFF threshold crossing, and event-wise reference updates that retain residual state. By separating the input trajectory from sensor-side conversion, FracEvent serves as a reusable event-generation core: different renderers, interpolation modules, or frame sources can provide trajectories while the same pixel-dynamics model handles event timing, polarity, and residual memory.

Experiments support FracEvent as a practical event-generation core. Under matched inputs, event-stream comparisons show that FracEvent better preserves repeated-trigger timing and produces spatiotemporal event patterns closer to real streams. Downstream tests provide complementary evidence under fixed protocols: models trained on GoPro-derived events generated by FracEvent achieve the lowest MSE and LPIPS on DAVIS240C and HQF reconstruction, and the optical-flow model trained on events generated by FracEvent is closest to the Real MVSEC control by mean AEE. The residual gap to real events clarifies the scope of this result: better sensor-side dynamics improve timing, polarity, and transfer, but they cannot compensate for all errors in the supplied brightness trajectory. Interpolation, rendering, exposure handling, calibration, and camera-specific effects therefore remain part of the simulation problem. Future work can pair this compact core with stronger trajectory models and camera mechanisms, and test broader tasks, sensors, and noise regimes.

References

- [1] Patrick Bardow, Andrew J. Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 884–892, 2016.
- [2] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240 x 180 130 db 3 us latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014.
- [3] Kenneth S. Cole and Robert H. Cole. Dispersion and absorption in dielectrics i. alternating current characteristics. *The Journal of Chemical Physics*, 9(4):341–351, 1941.
- [4] Burak Ercan, Onur Eker, Aykut Erdem, and Erkut Erdem. EVREAL: Towards a comprehensive benchmark and analysis suite for event-based video reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 3943–3952. IEEE, 2023.
- [5] Gereon Fox, Xingang Pan, Ayush Tewari, Mohamed Elgharib, and Christian Theobalt. Unsupervised event-based video reconstruction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4179–4188, 2024.
- [6] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jorg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022.
- [7] Mathias Gehrig, Daniel Gehrig, Javier Hidalgo-Carrio, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3583–3592, 2020.
- [8] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. DSEC: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6(3): 4947–4954, 2021.
- [9] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-RAFT: Dense optical flow from event cameras. In *Proceedings of the International Conference on 3D Vision*, pages 197–206, 2021.
- [10] Daxin Gu, Jia Li, Lin Zhu, Yu Zhang, and Jimmy S. Ren. Reliable event generation with invertible conditional normalizing flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2):927–943, 2024.
- [11] Haiqian Han, Jiacheng Lyu, Jianing Li, Henglu Wei, Cheng Li, Yajing Wei, Shu Chen, and Xiangyang Ji. Physical-based event camera simulator. In *Computer Vision – ECCV 2024*, pages 19–35. Springer, 2024.
- [12] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From video frames to realistic DVS events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1312–1321, 2021.
- [13] Yixuan Hu, Yuxuan Xue, Simon Klenk, Daniel Cremers, and Gerard Pons-Moll. ControlEvents: Controllable synthesis of event camera data with foundational prior from image diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5509–5519, 2026.
- [14] Khadija Iddrisu, Waseem Shariff, Suzanne Little, and Noel O’Connor. SynSacc: A blender-to-V2E pipeline for synthetic neuromorphic eye-movement data and sim-to-real spiking model training. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 675–684, 2026.
- [15] Xiao Jiang, Fei Zhou, and Jiongzhi Lin. ADV2E: Bridging the gap between analogue circuit and discrete frames in the video-to-events simulator. arXiv preprint arXiv:2411.12250, 2024.
- [16] Pritam P. Karmokar, Quan H. Nguyen, and William J. Beksi. Secrets of edge-informed contrast maximization for event-based vision. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 630–639, 2025.
- [17] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E. Shi, and Ryad B. Benosman. HOTS: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1346–1359, 2017.
- [18] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128 x 128 120 db 15 us latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008.
- [19] Songnan Lin, Ye Ma, Zhenhua Guo, and Bihan Wen. DVS-Voltmeter: Stochastic process-based event simulator for dynamic vision sensors. In *Computer Vision – ECCV 2022*, pages 578–593. Springer Nature Switzerland, 2022.
- [20] Francesco Mainardi. *Fractional Calculus and Waves in Linear Viscoelasticity: An Introduction to Mathematical Models*. Imperial College Press, London, 2010.
- [21] Ana I. Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5419–5427, 2018.
- [22] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *The International Journal of Robotics Research*, 36(2):142–149, 2017.
- [23] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3883–3891, 2017.
- [24] Zijie Ning, Enmin Lin, Sudarshan R. Iyengar, and Patrick Vandewalle. Raw2Event: Converting raw frame camera into event camera. arXiv preprint arXiv:2509.06767, 2025.
- [25] Gintautas Palinauskas, Camilo Amaya, Evan Eames, Michael Neumeier, and Axel von Arnim. Generating event-based datasets for robotic applications using MuJoCo-ESIM. In *Proceedings of the 2023 International Conference on Neuromorphic Systems*, pages 1–7. ACM, 2023.

- [26] Igor Podlubny. *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*. Academic Press, San Diego, 1999.
- [27] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. ESIM: An open event camera simulator. In *Proceedings of The 2nd Conference on Robot Learning*, pages 969–982. PMLR, 2018.
- [28] Henri Rebecq, Rene Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3857–3866, 2019.
- [29] Thilo Reinold, Suman Ghosh, and Guillermo Gallego. Combined physics and event camera simulator for slip detection. In *Proceedings of the Winter Conference on Applications of Computer Vision Workshops*, pages 935–943, 2025.
- [30] Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Fast image reconstruction with an event camera. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 156–163, 2020.
- [31] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. HATS: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1731–1740, 2018.
- [32] Timo Stoffregen, Cedric Scheerlinck, Davide Scaramuzza, Tom Drummond, Nick Barnes, Lindsay Kleeman, and Robert Mahony. Reducing the sim-to-real gap for event cameras. In *European Conference on Computer Vision*, pages 534–549. Springer, 2020.
- [33] Cedric Villani. *Optimal Transport: Old and New*. Springer, Berlin, Heidelberg, 2009.
- [34] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [35] Chuanzhi Xu, Langyi Chen, Haodong Chen, Vera Chung, and Vincent Qu. Towards end-to-end neuromorphic voxel-based 3D object reconstruction without physical priors. In *2025 IEEE International Conference on Multimedia and Expo*, pages 1–6, 2025.
- [36] Chuanzhi Xu, Haoxian Zhou, Langyi Chen, Haodong Chen, Zeke Zexi Hu, Zhicheng Lu, Ying Zhou, Vera Chung, Qiang Qu, and Weidong Cai. A survey of 3D reconstruction with event cameras. *arXiv preprint arXiv:2505.08438*, 2025.
- [37] Chuanzhi Xu, Haoxian Zhou, Langyi Chen, Yuk Ying Chung, and Qiang Qu. Ultralight polarity-split neuromorphic SNN for event-stream super-resolution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(13): 11196–11204, 2026.
- [38] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.
- [39] Zhongyang Zhang, Shuyang Cui, Kaidong Chai, Haowen Yu, Subhasis Dasgupta, Upal Mahbub, and Tauhidur Rahman. V2CE: Video to continuous events simulator. In *2024 IEEE International Conference on Robotics and Automation*, pages 12455–12461, 2024.
- [40] Haoxian Zhou, Chuanzhi Xu, Langyi Chen, Pengfei Ye, Haodong Chen, Yuk Ying Chung, and Qiang Qu. Exploiting spatiotemporal properties for efficient event-driven human pose estimation. *arXiv preprint arXiv:2512.06306*, 2025.
- [41] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based visual inertial odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5391–5399, 2017.
- [42] Alex Zihao Zhu, Dinesh Thakur, Tolga Ozaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multi vehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039, 2018.
- [43] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In *Robotics: Science and Systems*, 2018.
- [44] Alex Zihao Zhu, Ziyun Wang, Kaung Khant, and Kostas Daniilidis. EventGAN: Leveraging large scale image datasets for event cameras. In *2021 IEEE International Conference on Computational Photography*, pages 1–11, 2021.
- [45] Andreas Ziegler, Daniel Teigland, Jonas Tebbe, Thomas Gossard, and Andreas Zell. Real-time event simulation with frame-based cameras. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11669–11675, 2023.

Supplementary Material

Supplement Scope

In this supplementary material, we provide additional information on input and event-format conventions, dataset roles and splits, hardware and software environment, sensor-family parameter calibration, event-stream comparison details, image reconstruction metrics, qualitative reconstruction examples, and optical-flow estimation details. The material is organized around input conventions and experiment details: input and event-format conventions; dataset roles, splits, and environment; parameter settings and calibration; event-stream comparison details; image reconstruction details; and optical-flow estimation details.

A. Input and Event-Format Conventions

Scope. The method starts from a supplied intensity trajectory and outputs event tuples. This section records the input and event-format conventions used across the reported rows, so that comparisons isolate sensor-side event dynamics rather than upstream trajectory handling or event encoding.

Input Normalization and Trajectory Boundary. All event-generation rows start from grayscale intensities normalized to a common range before the log transform. The simulator then applies the same dark-current floor and numerical epsilon. Consequently, gamma handling, raw demosaicing, exposure reconstruction, color-to-grayscale conversion, and high-dynamic-range merging remain outside the simulator parameter set. The reported comparisons therefore fix the input pipeline at normalized grayscale trajectories and evaluate the sensor-side fractional-memory dynamics under those supplied trajectories. Interpolation, rendering, exposure recovery, raw-camera processing, and target-frame spacing are likewise treated as upstream trajectory choices.

Timestamp Convention. All simulator timestamps are in seconds. Frame timestamps, generated event timestamps, MVSEC frame intervals, and reconstruction event windows use the same unit before training code consumes the events. A millisecond-versus-second mismatch changes event-window duration by a factor of one thousand while leaving array shapes superficially valid. Timestamp consistency is therefore part of the event-source definition; rows are comparable only when this convention is preserved.

Coordinate and Polarity Convention. The raw event tuple uses horizontal coordinate x , vertical coordinate y , timestamp t , and polarity p . Polarity is encoded as $+1$ for ON and -1 for OFF. The OFF threshold parameter is stored as a positive magnitude and is applied through the signed residual condition $u(t) \leq -\theta_{\text{off}}$. Downstream event images preserve this convention when constructing positive and negative channels. If a loader swaps axes, flips signs, or interprets coordinates as rows and columns without conversion, the row no longer measures the intended event-source difference.

B. Dataset Roles, Splits, and Environment

Dataset-Role Ledger. The datasets in the main paper have distinct evidence roles. Some provide real events for sensor-family calibration or event-stream comparison, some provide high-frame-rate frames for synthetic generation, and some provide real-event targets for downstream transfer. This ledger keeps generation sources, event-statistics calibration and diagnostics, and downstream task evidence explicit. When a dataset supports more than one role, calibration uses only event-level statistics from specified windows and does not use downstream losses, target annotations, or validation metrics as feedback.

DAVIS240C Role. DAVIS240C is used in three distinct ways [1, 7]. First, selected DAVIS240C sequences provide APS frames and real events for selecting DAVIS240-series sensor-family parameter settings. Second, separate held-out DAVIS240C windows are used for event-stream comparison. Third, full DAVIS240C reconstruction targets are used to test whether a model trained from synthetic GoPro events transfers to real events. The parameter-selection windows and holdout-signature windows are separate, so the signature table is not a calibration objective reprinted as a result.

DAVIS346 real capture role. The DAVIS346 real capture is a small self-collected real-event sample used as additional real-event validation in the main paper. It provides qualitative event-stream comparison in real-life scenes and supports the DAVIS346-series sensor-regime discussion, without being used to train the downstream optical-flow model or select parameters from flow labels.

GoPro role. GoPro provides high-frame-rate intensity input for the reconstruction training source [8]. Its role is to supply a common brightness trajectory from which FracEvent, v2e [4], ESIM [3, 9], and DVS-Voltmeter [6] can each generate training streams. The reconstruction table can therefore be read as a matched-input comparison of event sources rather than a comparison between unrelated scene collections. Trajectory variants are upstream sensitivity checks and should remain separate from the matched event-source comparison.

HQF role. HQF is used as an additional real-event reconstruction target [12]. It complements DAVIS240C by evaluating the same matched reconstruction protocol on different scenes, exposure statistics, motion patterns, and evaluation frames. Parameter selection uses DAVIS240C event-level signatures, while HQF provides an independent cross-dataset transfer check. The HQF results are therefore reported as complementary real-event reconstruction evidence under the matched GoPro training setup.

MVSEC role. MVSEC supplies the optical-flow downstream lane [16]. The protocol trains each source on `outdoor_day2` and evaluates on `outdoor_day1` and the indoor flying sequences. MVSEC also provides DAVIS346-series event-statistics windows for parameter calibration; this calibration use is restricted to event-level statistics and does not use flow labels or EV-FlowNet validation feedback. This lane is independent from reconstruction: it uses the DAVIS346-series sensor-family setting, a different event representation, a different downstream model, and a different metric family. It tests whether the synthetic event stream supports motion learning under a fixed self-supervised flow protocol. Table S1 collects these dataset roles and feedback boundaries.

Table S1. **Dataset roles in the reported evidence.** Downstream lanes provide transfer evidence, while calibration and diagnostics document protocol controls.

Dataset	Paper role	Primary output	Boundary
DAVIS240C	Calibration, held-out signatures, real reconstruction target	Sensor-family setting, signature table, DAVIS240C metrics	Calibration windows are separated from holdout signatures.
DAVIS346 real capture	Self-collected additional real-event validation	Real-life event-stream visual evidence	Supports sensor-regime discussion without flow-label feedback.
GoPro	Common frame source for synthetic training events	Matched reconstruction simulator rows	Frame-source role; upstream trajectory variants are separate protocol checks.
HQF	Additional real-event reconstruction target	Full real event reconstruction metrics	Cross-dataset reconstruction target.
MVSEC	DAVIS346-series calibration and optical-flow transfer benchmark	Sensor-family setting, EV-FlowNet-style AEE and outlier table	Calibration uses event statistics; flow evaluation uses fixed protocol.

Execution environment. Table S2 documents the workstation and software environment used for the reported runs. Hardware is included for reproducibility context only; it is not a scientific variable in the comparisons.

Table S2. **Execution environment.** The table lists the workstation and software stack used for the reported experiments.

Item	Value
CPU	Intel Core i5-14600KF
GPU	NVIDIA RTX 5070 Ti, 16 GB
Python	3.11.15
PyTorch	2.11.0+cu130
CUDA	13.0

C. Parameter Settings and Calibration

Calibration role. Event-camera parameters are sensor- and protocol-dependent. The paper uses separate parameter settings for the DAVIS240-series and DAVIS346-series sensor families. These settings are calibrated from real event-stream statistics for the corresponding sensor family, not from image-reconstruction or optical-flow performance. The calibration objective

combines event count, temporal histogram structure, polarity behavior, and time-surface similarity [5, 11]. This makes the selected settings auditable as event-statistics fits rather than single-count adjustments. The same objective is not applied to ESIM, v2e, or DVS-Voltmeter; those baselines use their released calibrated or recommended settings so that the comparison does not impose a FracEvent-specific tuning rule on competing simulators.

Calibration objective. For each candidate parameter setting, the calibration pass compares generated and real event streams on calibration windows with the following weighted score:

$$\mathcal{J} = e_{\text{count}} + 0.5 e_{\text{temp}} + 0.35 e_{\text{pol}} + 0.25 (1 - \max(c_{\text{surf}}, 0)). \quad (\text{S1})$$

Here e_{count} is a mean absolute log-count error, e_{temp} is a temporal-histogram distance, e_{pol} measures polarity imbalance, and c_{surf} is a time-surface correlation. Count matching receives the largest weight because a generator with a large density error can still look plausible in normalized plots. Timing, polarity, and recency-surface terms keep parameter selection from collapsing to count matching alone. For DAVIS346-series calibration, the same local terms are combined with an MVSEC `outdoor_day2` event-density term so the selected setting also matches DAVIS346-series real-event density on the training split. Table S3 summarizes these objective terms and weights.

Table S3. **Calibration objective terms.** The selected setting minimizes a weighted event-statistics objective.

Term	Weight	Interpretation
Log-count error	1.00	Penalizes large event-density mismatch across windows.
Temporal histogram L1	0.50	Checks whether events occupy similar temporal portions of the window.
Polarity error	0.35	Discourages ON/OFF imbalance that can be hidden by total count.
Time-surface penalty	0.25	Rewards spatial-temporal recency agreement without making it the only target.

Parameter selection and holdout use. Using this local objective, the DAVIS240-series search evaluates 240 calibrated sensor-parameter candidates on DAVIS240C windows and selects $\alpha = 0.76$, $M = 4$, $\tau_{\text{ref}} = 0.0025$ s, $\theta_{\text{on}} = 0.5625$, and $\theta_{\text{off}} = 0.4000$. Using the same local terms with the event-density term, the DAVIS346-series search evaluates 588 candidates on 20 activity-stratified MVSEC `outdoor_day2` windows and selects $\alpha = 0.72$, $M = 6$, $\tau_{\text{ref}} = 0.0025$ s, $\theta_{\text{on}} = 0.4500$, and $\theta_{\text{off}} = 0.5000$. The reported DAVIS240C rows use the DAVIS240-series setting, while the reported MVSEC rows use the DAVIS346-series setting. The checks summarized in Table S4 are calibration checks, not downstream task results.

Calibration windows and feedback boundary. The DAVIS240C fine search uses four calibration sequences: `boxes_6dof`, `dynamic_rotation`, `poster_translation`, and `slider_depth`. Each candidate is evaluated on the first 40 APS frames of those calibration sequences. The holdout signature check uses separate windows from `boxes_rotation`, `dynamic_translation`, `poster_rotation`, and `slider_far`, with the first 80 APS frames per sequence. The DAVIS346-series search uses 20 activity-stratified windows from MVSEC `outdoor_day2` to match real-event density and local event statistics. The full downstream reconstruction table is then trained from GoPro-generated events and evaluated on full DAVIS240C and HQF targets. The full downstream optical-flow table is trained and evaluated with the fixed MVSEC protocol described in Section F. Parameter selection is therefore tied to event-level signatures rather than reconstruction losses, optical-flow labels, EV-FlowNet feedback, or downstream validation metrics.

Calibration search scope. The search selects a finite parameter setting for the reported sensor family and protocol. A different camera, bias setting, lens, exposure regime, or upstream trajectory source could require a different setting. The selected values should therefore be read as fixed sensor-family settings used for DAVIS240C and MVSEC rows, analogous to calibrated sensor parameters used by simulator baselines. Held-out DAVIS240C windows serve as event-level signature checks after the setting is fixed.

Fixed implementation settings. In addition to the five searched sensor-family parameters in Table S4, FracEvent uses fixed implementation settings shared by both reported profiles. Table S5 lists implementation constants and input conventions rather than dataset-specific calibrated degrees of freedom.

Table S4. **Sensor-family parameter settings.** Both settings are calibrated from real event-stream statistics and fixed before downstream evaluation; the table lists the calibration source, search scope, and selected parameters.

Item	DAVIS240-series	DAVIS346-series
Calibration source	DAVIS240C windows	MVSEC windows
Window plan	Four sequences, first 40 APS frames	20 <code>outdoor_day2</code> windows for event-level statistics matching
Candidates evaluated	240	588
α and modes M	0.76, 4	0.72, 6
τ_{ref}	0.0025 s	0.0025 s
θ_{on} and θ_{off}	0.5625, 0.4000	0.4500, 0.5000

Table S5. **Fixed implementation settings for FracEvent.** The listed values are shared across the reported DAVIS240-series and DAVIS346-series profiles unless otherwise stated.

Setting	Value	Role
Relaxation log range R	3.0	Samples mode offsets $u_m \in [-R, R]$ for $\tau_m = \tau_0 e^{u_m}$.
$\tau_{\text{min}}, \tau_{\text{max}}$	5×10^{-5} s, 3.0×10^{-2} s	Bounds the brightness-dependent base time constant.
I_{ref}, β	0.5, 1.0	Sets the brightness dependence of the base time constant.
$i_{\text{dark}}, \epsilon$	$0.001, 10^{-6}$	Stabilizes log-intensity normalization near zero intensity.
Gain g	1.0	Converts voltage differences to threshold units.
Bisection steps	12	Root-localization iterations for counted threshold levels.
Solver cap, minimum split step	64 events per interval, 10^{-6} s	Controls recursive splitting in dense intervals.
Input convention	Normalized grayscale, seconds, piecewise-linear $L(t)$	Shared timestamp, intensity, and interpolation convention.

Sensor-family setting separation. The reported datasets are captured by different DAVIS sensor families. DAVIS240C rows use a DAVIS240-series setting, while MVSEC rows use a DAVIS346-series setting calibrated from MVSEC event-stream windows. Reusing a DAVIS240-series setting for DAVIS346-series data would mix two camera and protocol regimes. The DAVIS346-series setting is therefore selected separately, again without downstream labels or EV-FlowNet validation feedback.

Threshold-solver condition. The event-generation solver counts threshold levels spanned by the endpoint voltage and then localizes each counted level by bisection. This root-localization step is exact for a pixel on a processed subinterval where the weighted voltage trajectory is monotone, because each counted threshold level has a unique crossing. The carried fractional mode state can make a long interval’s weighted voltage non-monotone, so the implementation does not claim a global monotonicity proof. Instead, dense intervals are recursively split before bisection, which keeps the endpoint-count assumption local while avoiding a uniform substep grid.

D. Event-Stream Comparison Details

Matched-window protocol. The DAVIS240C event-stream comparison uses the four held-out sequences listed in Section C. For each sequence, eight deterministic frame-aligned 50 ms windows are sampled from the first 80 APS frames, giving 32 matched windows in total. The windows are fixed before scoring and are shared by all simulator baselines and component ablations. All event sources are scored against the same real DAVIS240C windows with identical start and end timestamps, spatial resolution, timestamp units, polarity convention, and metric implementation. For the component ablation, the input frames, DAVIS240-series setting, matched windows, and metrics are fixed; only the named FracEvent component is removed. This protocol keeps the event-source dynamics as the intended variable and prevents window selection, timestamp alignment, or tensor construction from becoming hidden degrees of freedom. Figure S1 shows additional qualitative event-stream comparisons under this fixed-window protocol.

Metric details. For a simulated event window S and the matched real event window R , the count ratio is:

$$r_{\text{count}}(S, R) = \frac{|S| + 1}{|R| + 1}. \quad (\text{S2})$$

The additive one keeps empty or near-empty windows numerically defined.

Let $\mathcal{I}(E)$ denote the positive same-pixel consecutive intervals obtained after sorting an event stream E by pixel and

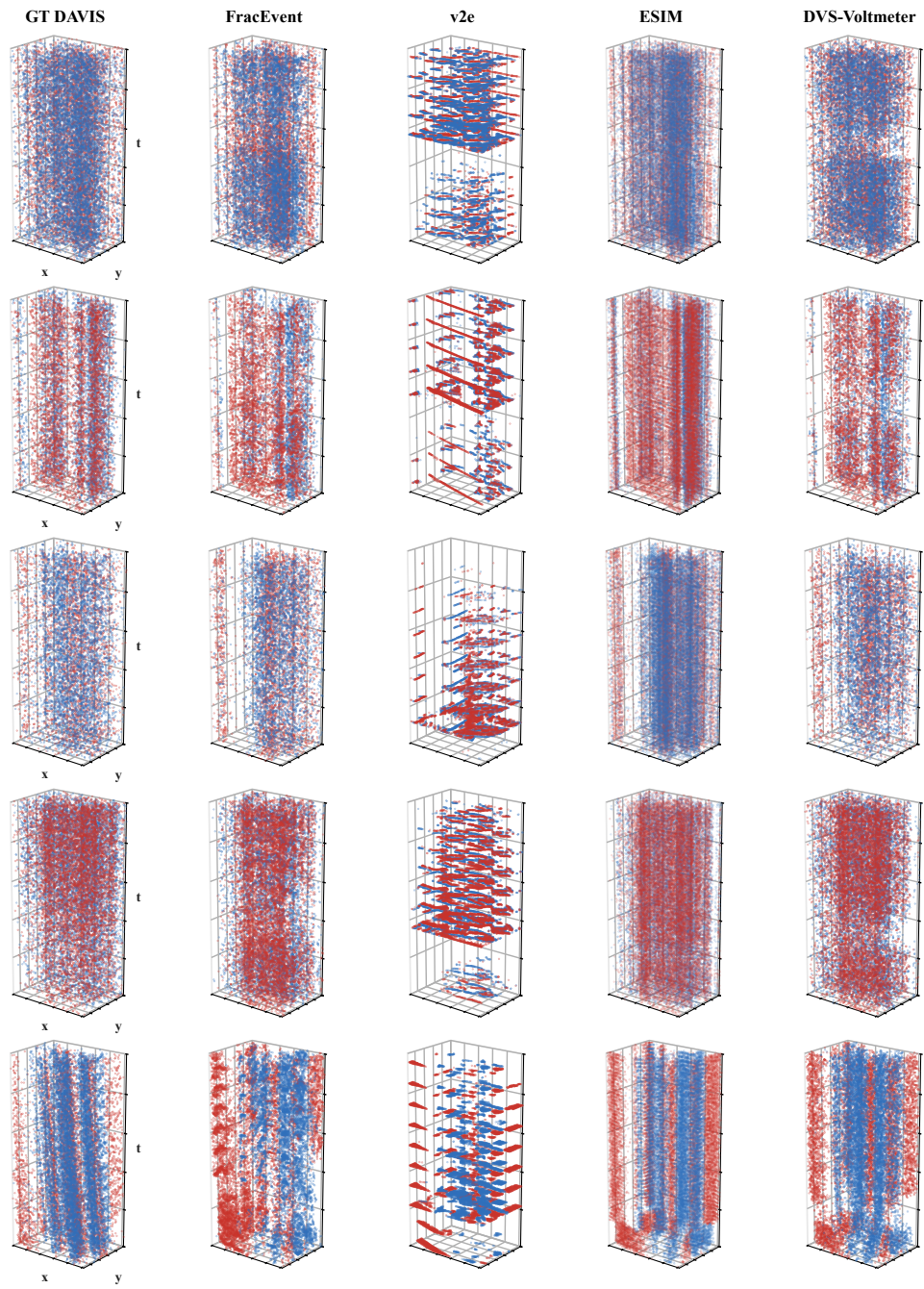


Figure S1. **Additional qualitative event-stream comparisons.** Each row shows a spatiotemporal event plot for one 50 ms DAVIS240C window, with columns comparing real events and generated events from the same interval. Red and blue denote ON and OFF events.

timestamp. The IEI distance clips intervals below 10^{-6} s before the logarithm and applies the 1-Wasserstein distance [13]:

$$\begin{aligned}\tilde{\mathcal{I}}(E) &= \log_{10}(\max(\mathcal{I}(E), 10^{-6})), \\ d_{\text{IEI}}(S, R) &= W_1(\tilde{\mathcal{I}}(S), \tilde{\mathcal{I}}(R)).\end{aligned}\tag{S3}$$

Windows without valid same-pixel intervals are excluded from the IEI aggregate for that metric rather than assigned an artificial interval value.

ON/OFF balance is the absolute difference between simulated and real ON-event fractions:

$$\begin{aligned}\pi_+(E) &= \begin{cases} |E|^{-1} \sum_{e \in E} \mathbf{1}[p_e = +1], & |E| > 0, \\ 0.5, & |E| = 0, \end{cases} \\ d_{\text{pol}}(S, R) &= |\pi_+(S) - \pi_+(R)|.\end{aligned}\tag{S4}$$

For the time-surface metric [5, 11], let $t_p^E(x, y)$ be the latest timestamp of polarity p at pixel (x, y) in stream E , and set the corresponding surface entry to zero when no such event exists. Using $\tau = 0.05$ s and the common window end time t_1 , we compute:

$$\begin{aligned}T_p^E(x, y) &= \exp\left(-\frac{t_1 - t_p^E(x, y)}{\tau}\right), \\ d_{\text{surf}}(S, R) &= 1 - \max\left(\frac{1}{|\mathcal{P}_v|} \sum_{p \in \mathcal{P}_v} \rho(T_p^S, T_p^R), 0\right),\end{aligned}\tag{S5}$$

where \mathcal{P}_v contains polarities with finite correlations. Together, these metrics measure density, repeated-trigger timing, polarity balance, and recent-event spatial structure under one fixed window protocol.

Repeated-trigger real-event diagnostics. Real event streams contain frequent same-pixel repeats, with capture-dependent variation. This audit is a real-event mechanism check rather than an additional simulator ranking; it characterizes repeated-trigger and short-interval burst structure in real event streams. These measurements provide context for the retained-memory and continuous-time crossing behavior evaluated by the main-paper event-stream and ablation tables. The analysis samples 16 deterministic, uniformly spaced 50 ms windows per capture and reports median [IQR] percentages across windows, where IQR is the 25th–75th percentile range. Repeated-pixel ratio is the fraction of active pixels that fire at least twice in the window. Burst-IEI ratio is the fraction of same-pixel consecutive event intervals below 10 ms. Multi-event contribution is the fraction of all events emitted by pixels that fire at least twice in the same window. Table S6 reports sampled real-event evidence across DAVIS240C and MVSEC captures.

Table S6. **Per-capture repeated-trigger analysis.** Entries are median [IQR] percentages over sampled 50 ms real-event windows; IQR is the 25th–75th percentile range.

Capture	Win.	Rep. px	IEI < 10ms	Rep. events
DAVIS240C boxes_rotation	16	89.6 [68.8, 95.1]	73.1 [62.8, 79.8]	97.7 [87.2, 99.1]
DAVIS240C dynamic_translation	16	62.2 [54.9, 65.5]	71.4 [69.9, 73.7]	83.9 [78.4, 86.3]
DAVIS240C poster_rotation	16	83.2 [65.5, 95.4]	72.8 [63.3, 79.8]	94.6 [85.9, 99.1]
DAVIS240C slider_far	16	41.8 [37.1, 47.0]	51.9 [46.5, 57.3]	64.3 [59.5, 69.9]
MVSEC indoor_flying1	16	10.3 [8.6, 15.5]	0.4 [0.2, 1.3]	19.2 [16.6, 27.5]
MVSEC indoor_flying2	16	13.7 [10.1, 17.0]	0.4 [0.2, 0.9]	24.7 [19.1, 29.8]
MVSEC indoor_flying3	16	12.8 [10.8, 14.6]	0.4 [0.2, 0.5]	23.3 [20.2, 26.2]
MVSEC outdoor_day1	16	39.4 [32.0, 46.6]	61.1 [57.2, 63.2]	64.9 [55.1, 70.9]
MVSEC outdoor_day2	16	50.0 [36.6, 62.4]	59.9 [57.2, 67.5]	74.0 [59.0, 85.0]

E. Image Reconstruction Details

Protocol isolation. The reconstruction lane trains the same E2VID-style recurrent model [10] from synthetic GoPro events and evaluates the resulting trained model on real event streams. All synthetic rows share the same GoPro source, 240×180 training resolution, target-frame construction, loss, and optimizer budget. The test sets are full DAVIS240C and full HQF sequences, not short visual excerpts.

Evaluation granularity. The evaluation averages over the full target sequences: 22,680 DAVIS240C frames and 15,498 HQF frames in the matched GoPro comparison. Full-sequence averaging matters because qualitative reconstructions can vary sharply with scene texture, illumination, and motion.

Reconstruction metrics. Let $I_t, \hat{I}_t \in [0, 1]^N$ denote the target and reconstructed grayscale frames over N valid pixels. The main reconstruction table reports full-sequence averages of MSE, SSIM, and LPIPS, following event-reconstruction reporting practice [2]. The pixelwise error metric is:

$$\text{MSE}_t = \frac{1}{N} \sum_{i=1}^N (\hat{I}_{t,i} - I_{t,i})^2. \quad (\text{S6})$$

SSIM measures luminance, contrast, and structural agreement [14]:

$$\text{SSIM}(\hat{I}, I) = \frac{(2\mu_{\hat{I}}\mu_I + C_1)(2\sigma_{\hat{I}I} + C_2)}{(\mu_{\hat{I}}^2 + \mu_I^2 + C_1)(\sigma_{\hat{I}}^2 + \sigma_I^2 + C_2)}. \quad (\text{S7})$$

LPIPS compares learned deep features with calibrated channel weights [15]:

$$\text{LPIPS}(\hat{I}, I) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| w_l \odot (\phi_l(\hat{I})_{hw} - \phi_l(I)_{hw}) \right\|_2^2. \quad (\text{S8})$$

Lower MSE and LPIPS indicate better agreement, while higher SSIM indicates stronger structural similarity.

Event windows and target alignment. The reconstruction learner receives event tensors built from raw event streams and learns to predict intensity targets at the paired frame timestamps. This pairing is part of the protocol: changing the number of frames between targets changes the temporal integration problem even if the same sensor-family setting is used. For this reason, the main simulator comparison fixes the source split, target-frame construction, spatial resolution, optimizer budget, and evaluation code. Rows that change target spacing or the event-window distribution are outside the primary simulator comparison.

Primary reconstruction target. The image-reconstruction comparison uses the DAVIS240-series setting because its primary real-event target is DAVIS240C. In this setting, DAVIS240C is the closest downstream target for asking whether the synthetic GoPro events transfer to the intended sensor regime. HQF adds an additional real-event target with different scenes, exposure statistics, motion patterns, and evaluation frames. Together, DAVIS240C and HQF evaluate matched-GoPro reconstruction transfer across primary and cross-dataset real-event targets. The main paper reports the metric-specific results; this supplement provides the corresponding protocol details and additional qualitative examples.

Qualitative reconstruction examples. Figures S2 and S3 broaden the visual comparison to additional DAVIS240C and HQF scenes under the same trained models and raw-display convention.

Brightness guardrail. The Figure 4 panels show raw $[0, 1]$ reconstruction outputs without per-method contrast normalization. Table S7 reports brightness statistics for the target and reconstructed outputs. These statistics confirm that brighter outputs do not, by themselves, indicate closer target agreement, which is why the main table reports MSE, SSIM, and LPIPS over full sequences.

Table S7. **Brightness diagnostics for the Figure 4 examples.** The main-paper reconstruction figure shows raw $[0, 1]$ outputs without per-method contrast normalization.

Source	Mean	P95	White	Black
Target	0.1038	0.3020	0.0000	0.3340
FracEvent	0.1005	0.2235	0.0000	0.2196
v2e	0.4424	0.9255	0.0381	0.0112
ESIM	0.3532	0.9647	0.0583	0.0001
DVS-Voltmeter	0.5359	0.8078	0.0014	0.0000



Figure S2. **Additional DAVIS240C reconstruction comparisons.** The panel extends the main reconstruction visual to more DAVIS240C scenes using the same matched GoPro-trained models, methods, metrics, and raw $[0, 1]$ display convention.



Figure S3. **Additional HQF reconstruction comparisons.** The panel shows the additional HQF real-event target under the same reconstruction protocol and display convention as the DAVIS240C comparison.

F. Optical-Flow Estimation Details

Optical-flow transfer role. Reconstruction evaluates whether synthetic events train a model to invert events into images. Optical flow evaluates a different property: whether event timing and motion structure support motion estimation. The MVSEC lane adds a motion-estimation check to the reconstruction lane. MVSEC provides synchronized event data, grayscale frames, and optical-flow ground truth for standard event-vision evaluation [16, 17].

EV-FlowNet-style protocol. We use an EV-FlowNet-style recipe [17] with fixed network architecture, event representation, losses, optimizer settings, augmentation, and 300k-step schedule. For the simulator comparison, each row trains one model on `outdoor_day2` and evaluates that same trained model on `outdoor_day1` official intervals and `indoor_flying1`, `indoor_flying2`, and `indoor_flying3` at $dt = 1$ and $dt = 4$. The `unified-outdoor_day2` split keeps the training event source as the controlled variable across outdoor and indoor tests. With this fixed EV-FlowNet-style recipe, transfer is measured consistently across all event sources.

Implementation alignment. Our implementation follows the released EV-FlowNet design in the components that affect the simulator comparison: four encoder stages, two residual blocks, decoder skip connections, multi-scale flow heads, Charbonnier photometric loss, image warping, and smoothness regularization. The input representation uses four event-image channels: positive counts, negative counts, latest positive timestamp, and latest negative timestamp. Training samples use random temporal gaps from the same gap set for every event source, and evaluation uses fixed frame skips. Grayscale APS images are kept in the official 0–255 range for the photometric loss, avoiding a normalization mismatch that would create a different training problem.

Real event control row. The real MVSEC row provides the reference scale for the fixed EV-FlowNet-style recipe when trained with real events from the same training split. With this reference row, the synthetic rows remain interpretable under the same split, model, and evaluation intervals. The full numerical comparison is reported in the main paper; this supplement gives the fixed protocol and additional qualitative examples.

Flow metrics. Following the EV-FlowNet-style reporting protocol [16, 17], average endpoint error is computed over valid optical-flow pixels as:

$$\text{AEE} = \frac{1}{|\Omega_v|} \sum_{(x,y) \in \Omega_v} \|\hat{\mathbf{u}}(x,y) - \mathbf{u}(x,y)\|_2, \quad (\text{S9})$$

where Ω_v is the set of pixels with valid MVSEC flow labels. The outlier percentage is:

$$\text{Out.} = \frac{100}{|\Omega_v|} \sum_{(x,y) \in \Omega_v} \mathbb{1}[e(x,y) > 3 \text{ px} \wedge e(x,y) > 0.05 \|\mathbf{u}(x,y)\|_2], \quad (\text{S10})$$

where $e(x,y) = \|\hat{\mathbf{u}}(x,y) - \mathbf{u}(x,y)\|_2$. We report both AEE and outlier percentage because AEE is sensitive to average error magnitude, while the outlier rate captures large-error regions. Table S8 lists the fixed training and evaluation controls used for every training event source.

DAVIS346-series sensor setting. The FracEvent optical-flow row uses the DAVIS346-series setting calibrated from activity-stratified MVSEC `outdoor_day2` event windows, with $\alpha = 0.72$, $M = 6$, $\tau_{\text{ref}} = 0.0025$ s, $\theta_{\text{on}} = 0.45$, and $\theta_{\text{off}} = 0.50$. The calibration objective matches real-event density while using local count, temporal histogram, polarity, and time-surface statistics. No flow labels or EV-FlowNet feedback are used for parameter selection. This separate setting reflects the DAVIS346-series sensor regime rather than downstream optical-flow tuning.

Sequence-level MVSEC reporting. The main paper reports AEE and outlier percentage for each MVSEC sequence at $dt = 1$ and $dt = 4$ under the `unified-outdoor_day2` simulator-comparison split described above. This section adds the protocol controls, calibration details, and qualitative MVSEC panels behind that compact table.

Table S8. **MVSEC optical-flow protocol controls.** These settings are fixed for every training event source.

Component	Fixed setting
Training split	<code>outdoor_day2</code> , left camera, one model per event source; the same trained model is used for outdoor and indoor evaluation.
Training length	300k optimizer steps for every row.
Batch and crop	Batch size 8 with 256×256 random crops.
Optimizer	Adam, learning rate 10^{-5} , weight decay 10^{-4} .
LR schedule	Multiplicative decay 0.8 every four epochs.
Event image	Positive count, negative count, latest positive time, latest negative time.
Image scale	APS grayscale images kept in 0–255 range for photometric loss.
Frame gaps	Shared random training gaps from {2, 4, 6, 8, 10, 12}; fixed $dt = 1$ and $dt = 4$ evaluation.
Augmentation	Random temporal gap, horizontal flip, and random crop; no random rotation.
Evaluation	<code>outdoor_day1</code> official interval plus <code>indoor_flying1</code> , <code>indoor_flying2</code> , and <code>indoor_flying3</code> , using the same <code>outdoor_day2</code> -trained model for all targets.
Metrics	Average endpoint error and outlier percentage.
Completion check	Full 300k-step training and all eight evaluation intervals were completed for each reported row.

Additional optical-flow examples. Figures S4 and S5 show additional MVSEC qualitative comparisons for the same fixed EV-FlowNet-style protocol. The panels use the same trained source rows, target sequences, metric definitions, and flow-color convention as the main optical-flow figure. The displayed intervals are `outdoor_day1` 10342–10346 and 10708–10712, `indoor_flying1` 1103–1107, and `indoor_flying2` 1333–1337; the corresponding 95th-percentile flow magnitudes are 4.36, 3.51, 6.23, and 16.33.

Interpreting event density. Event density gives useful context for the AEE and outlier results because it reports the scale of each training source. The real MVSEC control has 466.41M training events in this setup, and `FracEvent` closely matches that scale at 446.77M. `DVS-Voltmeter` is moderately denser at 535.01M, while `v2e` and `ESIM` are about $4.25\times$ and $11.85\times$ denser, respectively. This density context helps interpret the main MVSEC transfer result, where `FracEvent` is the closest synthetic source to the real event control by mean AEE under the fixed EV-FlowNet protocol.

MVSEC reproducibility details. Each MVSEC row uses a completed 300k-step training run, event-image tensors with the expected channel structure and dtype, and the same eight evaluation intervals. The reported table uses official-range APS images and completed 300k-step rows.

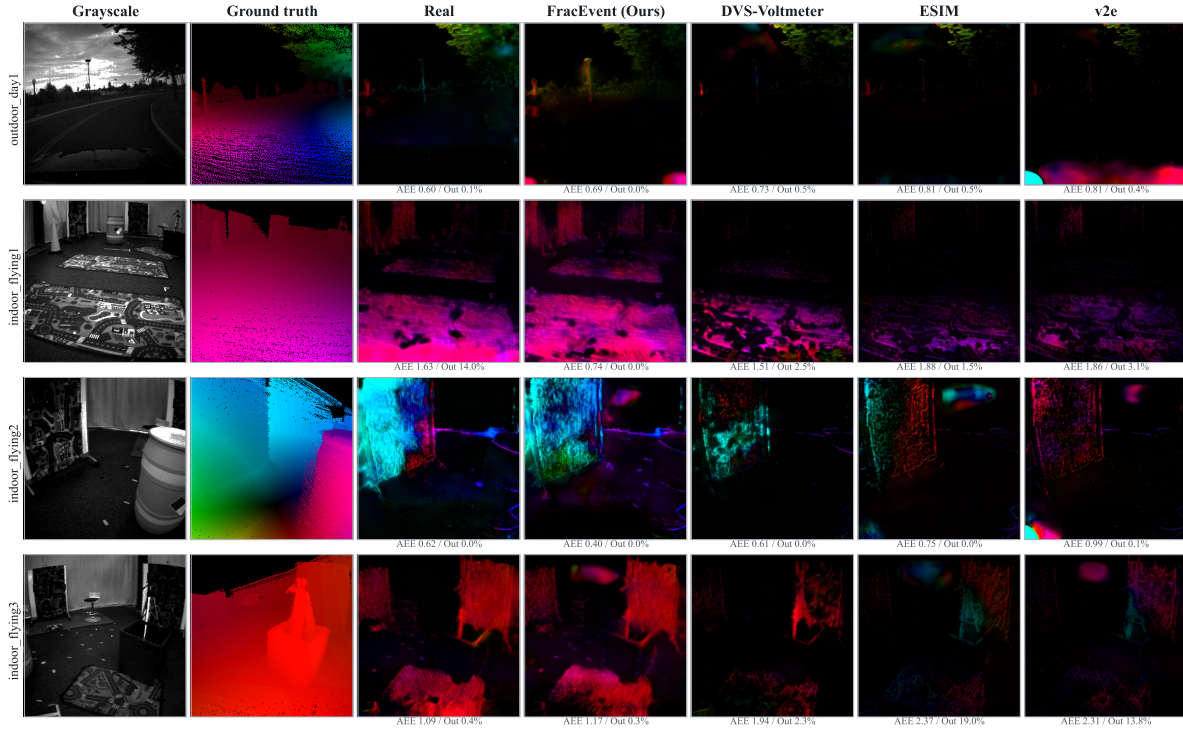


Figure S4. **Additional MVSEC optical-flow comparisons at $dt = 1$.** Rows show the evaluated MVSEC sequences; columns compare the real event control and synthetic training sources under the same optical-flow transfer protocol.

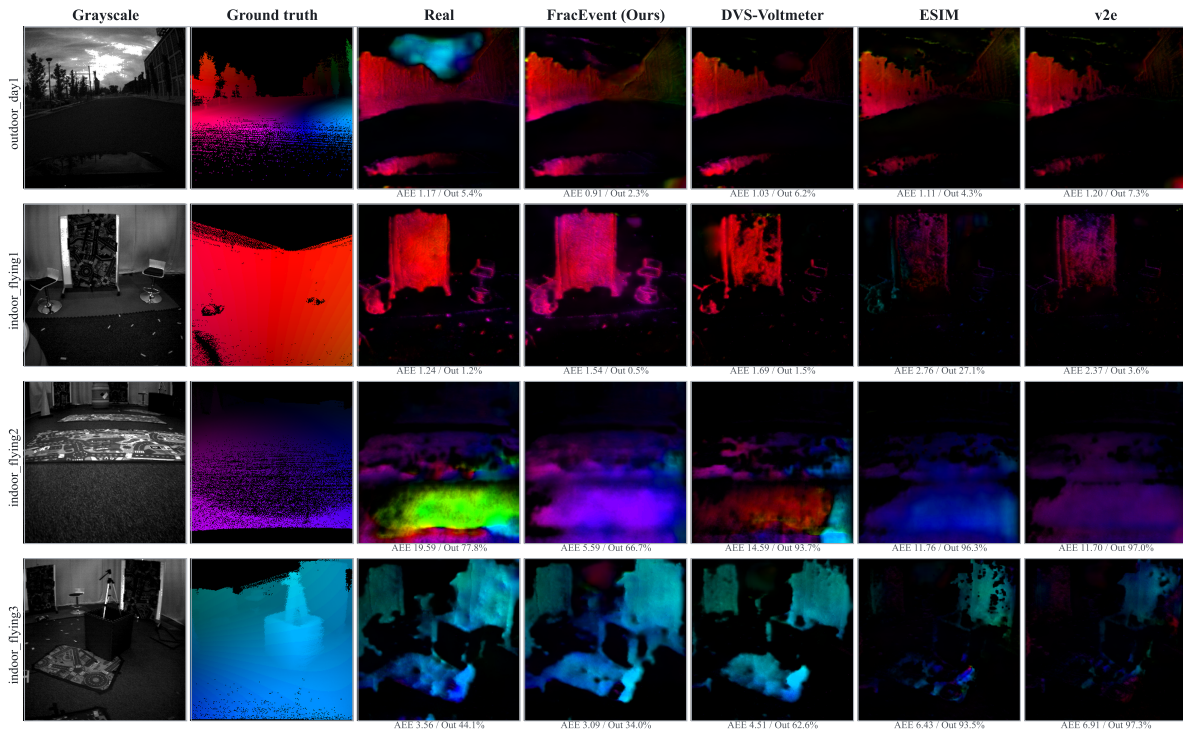


Figure S5. **Additional MVSEC optical-flow comparisons at $dt = 4$.** The panel complements the $dt = 1$ visualization with the longer-interval evaluation used in the reported MVSEC table.

References

- [1] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240 x 180 130 db 3 us latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014.
- [2] Burak Ercan, Onur Eker, Aykut Erdem, and Erkut Erdem. EVREAL: Towards a comprehensive benchmark and analysis suite for event-based video reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 3943–3952. IEEE, 2023.
- [3] Daniel Gehrig, Mathias Gehrig, Javier Hidalgo-Carrio, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3586–3595, 2020.
- [4] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From video frames to realistic DVS events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1312–1321, 2021.
- [5] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E. Shi, and Ryad B. Benosman. HOTS: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1346–1359, 2017.
- [6] Songnan Lin, Ye Ma, Zhenhua Guo, and Bihan Wen. DVS-Voltmeter: Stochastic process-based event simulator for dynamic vision sensors. In *Computer Vision – ECCV 2022*, pages 578–593. Springer Nature Switzerland, 2022.
- [7] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *The International Journal of Robotics Research*, 36(2):142–149, 2017.
- [8] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3883–3891, 2017.
- [9] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. ESIM: An open event camera simulator. In *Proceedings of The 2nd Conference on Robot Learning*, pages 969–982. PMLR, 2018.
- [10] Henri Rebecq, Rene Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3857–3866, 2019.
- [11] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. HATS: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1731–1740, 2018.
- [12] Timo Stoffregen, Cedric Scheerlinck, Davide Scaramuzza, Tom Drummond, Nick Barnes, Lindsay Kleeman, and Robert Mahony. Reducing the sim-to-real gap for event cameras. In *European Conference on Computer Vision*, pages 534–549. Springer, 2020.
- [13] Cedric Villani. *Optimal Transport: Old and New*. Springer, Berlin, Heidelberg, 2009.
- [14] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [15] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.
- [16] Alex Zihao Zhu, Dinesh Thakur, Tolga Ozaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multi vehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039, 2018.
- [17] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In *Robotics: Science and Systems*, 2018.