EDmamba: Rethinking Efficient Event Denoising with Spatiotemporal Decoupled SSMs

Ciyu Ruan[1,*], Zihang Gong[2,*], Ruishan Guo[1], Jingao Xu[3], Xinlei Chen[1,†]
[1]Tsinghua University, [2]Harbin Institute of Technology, [3]Carnegie Mellon University,
{softword77, gongzihang0201, ruishanguo314, xujingao13}@gmail.com,
chen.xinlei@sz.tsinghua.edu.cn

Abstract

Event cameras provide micro-second latency and broad dynamic range, yet their raw streams are marred by spatial artifacts (e.g., hot pixels) and temporally inconsistent background activity. Existing methods jointly process the entire 4D event volume (x, y, p, t), forcing heavy spatio-temporal attention that inflates parameters, FLOPs, and latency. We introduce EDmamba, a compact event-denoising framework that embraces the key insight that spatial and temporal noise arise from different physical mechanisms and can therefore be suppressed independently. A polarity- and geometry-aware encoder first extracts coarse cues, which are then routed to two lightweight state-space branches: a Spatial-SSM that learns location-conditioned filters to silence persistent artifacts, and a Temporal-SSM that models causal signal dynamics to eliminate bursty background events. This decoupled design distills the network to only 88.9K parameters and 2.27GFLOPs, enabling real-time throughput of 100K events in 68ms on a single GPU, 36× faster than recent Transformer baselines. Despite its economy, EDmamba establishes new state-of-the-art accuracy on four public benchmarks, outscoring the strongest prior model by 2.1 percentage points.

1 Introduction

Inspired by biological vision, event cameras asynchronously record per-pixel brightness changes with microsecond latency, ultra-high dynamic range (>120 dB), and low power consumption (<10 mW). These unique properties enable event-based perception to excel in high-speed and high-dynamic-range scenarios, powering breakthroughs in visual tracking [1], SLAM [2], and obstacle avoidance [3]. However, this high temporal resolution is a double-edged sword: it also amplifies sensitivity to minor brightness fluctuations, thermal noise, and sensor imperfections. As a result, event streams often contain a large number of spurious events that obscure valid motion patterns, hinder downstream perception, and overwhelm system bandwidth with excessive event rates [4]. Robust denoising is therefore a critical prerequisite for building scalable, high-performance event-driven systems.

Recent advances in event denoising have progressed from early methods based on statistical priors [5], spatiotemporal filtering [6], and surface fitting [7] to data-driven approaches. Deep learning-based models such as the CNN-based EDnCNN [8] and the point-based AEDNet [9] focus on local neighborhood patterns, while the Transformer-based EDformer [10] enables global context modeling but suffers from intensive computation. More recently, state space models like Mamba [11] have emerged for linear-time sequence modeling, and Pre-Mamba [12] extends this framework to 4D event deraining.

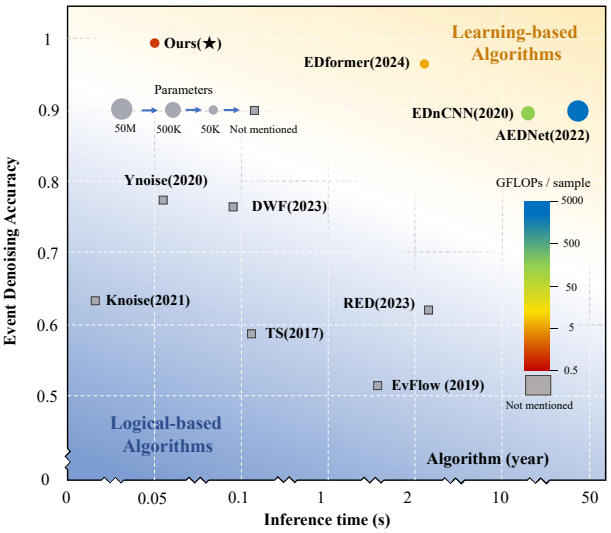

Figure 1: Performance vs. efficiency on event denoising with 100K events from DND21 (346×260, Hotel-bar and Driving scenes). EDmamba (red star) achieves state-of-the-art accuracy with low FLOPs, few parameters, and fast inference. All methods were evaluated under identical settings. Marker colors indicate FLOPs; sizes reflect parameter counts.

Despite architectural diversity, these models commonly treat event denoising as a 4D problem across spatial and temporal dimensions, and adopt unified spatio-temporal processing backbones. This joint modeling necessitates dense self- or cross-attention across space and time, resulting in redundant computation and limited adaptability to heterogeneous noise patterns. As a result, these models are often over-parameterized and suffer from high latency, limiting their suitability for real-time use. For instance, Transformer-based models can take over 2 seconds to process 100K events [10], which severely undermines the speed advantage of event cameras in high-throughput applications such as UAV navigation and autonomous driving.

To address this, we rethink event denoising from a noise-centric perspective. While event noise originates from diverse sources such as shot noise, fixed-pattern artifacts, and thermal leakage, its manifestations are often decoupled across time and space. Temporal noise, such as stochastic firings and polarity flips, lacks motion continuity and coherence, whereas spatial noise from hot pixels

* Equal contribution.
† Corresponding author.

produces localized, structurally aberrant activations. This observation motivates a decoupled architecture that separates spatial and temporal denoising into two lightweight, parallel branches. By isolating distinct noise patterns, this design reduces computational overhead and improves adaptability, offering lower latency without sacrificing accuracy.

Building on this insight, we propose **EDmamba**, a lightweight and effective **E**vent **D**enoising model that combines modality-specific processing with the efficiency of State Space Models. EDmamba operates directly on raw event streams represented as compact 4D event clouds, which encode spatial coordinates, polarity, and precise timestamps. A Coarse Feature Extraction (CFE) module first encodes geometric and polarity-aware features. These are then processed by two separate SSM branches: a Spatial Mamba (S-SSM) that captures local geometric patterns to suppress spatially incoherent noise, and a Temporal Mamba (T-SSM) that models temporal dynamics to eliminate temporally inconsistent activations. While structurally decoupled, the two streams interact via a shared Spatial-Temporal State Space Block (STSSB), enabling joint reasoning without entangled feature extraction. A U-Net-style encoder-decoder backbone ensures multi-scale information flow through skip connections, supporting both localized denoising and global context integration.

EDmamba achieves state-of-the-art performance with a compact design. It requires only 88.98K parameters and 2.27 GFLOPs, and efficiently processes 100K events in just 0.0685 seconds. Compared to recent Transformer-based methods, it achieves a 2.08% improvement in denoising accuracy while offering 36 times faster inference. This balance of accuracy, speed, and scalability makes EDmamba a practical and robust solution for real-time event-based perception. Our main contributions are as follows:

- We demonstrate that spatial and temporal noise in event streams exhibit distinct patterns and can be more effectively suppressed through decoupled denoising. This design insight enables simultaneous improvements in both accuracy and efficiency.
- We propose EDmamba, the first state space model specifically designed for event denoising. It extracts polarity- and geometry-aware features from 4D event clouds, and applies two lightweight, decoupled Mamba branches that independently model spatial and temporal noise characteristics for targeted suppression.
- We conduct extensive experiments demonstrating that EDmamba outperforms strong baselines in both denoising accuracy and inference speed, while requiring significantly fewer parameters.

2 Related Work

Event denoising has advanced through various approaches, including signal processing, statistical modeling, surface fitting, and deep learning. These methods have improved the robustness of event-based perception in noisy conditions.

Statistical methods. Early techniques leveraged statistical heuristics to suppress spurious events by evaluating event density within local spatiotemporal neighborhoods, rejecting low-density events as noise [5]. The pioneering work by Delbrück [13] proposed density-based filtering with spatial context. Subsequent improvements [14–16] optimized computational efficiency through enhanced event storage and processing. However, their reliance on manual parameter tuning hinders generalization across diverse scenarios.

Filtering-based methods. To better accommodate the sparse, asynchronous nature of event data, researchers have introduced filtering techniques along temporal, spatial, and spatiotemporal axes. Temporal filters [6] leverage temporal correlation, often exploiting patterns from edge motion. Spatial filters [17] analyze local intensity changes to distinguish signal events. Spatiotemporal filters [18, 19] integrate both domains to suppress background activity (BA) noise while preserving motion-related information. Notably, [14] showed that combining spatial and temporal cues yields more effective noise suppression than either alone.

Surface fitting techniques. Surface fitting offers an alternative denoising strategy by modeling the spatiotemporal distribution of events. EV-Gait [7] and GEF [20] apply local plane fitting and optical flow to differentiate noise from coherent motion. Time-surface (TS) representations [6, 21] convert event streams into decaying memory surfaces that encode temporal history, aiding in distinguishing structured signals from random outliers. These methods perform well under smooth motion but often degrade in fast dynamics or low-light environments.

Deep learning-based methods. Recent advances in deep learning have enabled data-driven event denoising through end-to-end learning. These models are typically trained on noisy-clean event pairs or self-supervised proxies. Early work like K-SVD [22] employed sparse feature learning, followed by EDnCNN [8], which fuses frame and IMU data via convolutional networks. EventZoom [23] introduced noise-to-noise training with a U-Net, and AEDNet [9] leveraged PointNet to process raw event streams. MLPF [16] explored probabilistic modeling, while Alkendi et al.[24] combined GNNs and transformers for per-event classification. EDformer[10] adopts a pure transformer architecture for event-wise denoising. Although not designed for denoising, Pre-Mamba [12] extends state space models to 4D event sequences for deraining, highlighting their potential for high-resolution event modeling.

3 Method

3.1 Working Principle

Our event denoising framework is grounded in the physical mechanisms of event generation and sensor noise. An event is triggered at pixel $\mathbf{u} = (x, y)^\top$ when the log-intensity change exceeds a contrast threshold C, yielding a polarity $p \in \{-1, +1\}$:

$$pC = \Delta L(\mathbf{u}, t) \triangleq \log I(\mathbf{u}, t) - \log I(\mathbf{u}, t - \Delta t). \tag{1}$$

This change can be decomposed as $\Delta L = \Delta L_s + \Delta L_n + \Delta L_c$, where ΔL_s denotes motion-induced signal, approximated by:

$$\Delta L_s \approx -\nabla \log I \cdot \mathbf{v} \Delta t, \tag{2}$$

where $\nabla \log I$ is the spatial log-intensity gradient and $\mathbf{v}$ the image-plane velocity. The remaining terms ΔL_n and ΔL_c denote photonic and circuit-level noise, respectively. Specifically, ΔL_n models photon shot noise and background activity, often causing temporally incoherent firings, while ΔL_c captures thermal leakage and fixed-pattern artifacts such as hot pixels, leading to spatially inconsistent

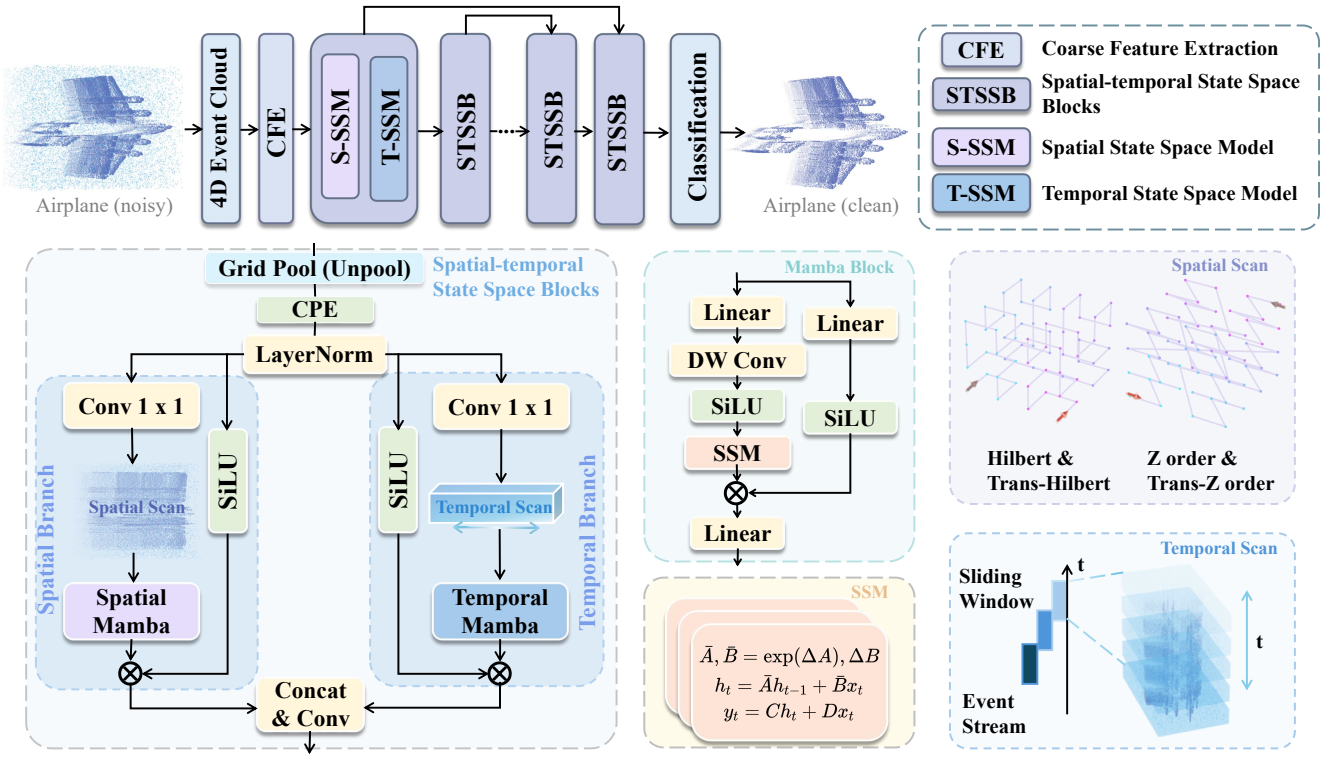

Figure 2: Overview of the EDmamba architecture. Raw events are grouped into 4D event clouds capturing spatial, temporal, and polarity information. The Coarse Feature Extraction (CFE) module projects events into geometric and polarity-aware subspaces. The U-Net-style denoising backbone employs multi-scale Spatial-Temporal State Space Blocks (STSSBs), composed of two complementary modules: Spatial SSM (S-SSM) suppresses spatially incoherent noise by modeling local geometric consistency, while Temporal SSM (T-SSM) filters temporally inconsistent events by capturing motion-aligned patterns across time.

activations:

$$\Delta L_n \sim \mathcal{N}(0, \sigma_n^2) + \lambda_{BA}\mathcal{P}(\gamma_{BA}),$$
$$\Delta L_c = \eta_{th}\left(\frac{k_B T}{e}\right) + \frac{I_{dark}\Delta t}{C_{pd}}, \tag{3}$$

where σ_n is the readout noise, λ_{BA} the background activity rate, γ_{BA} the gain, k_B the Boltzmann constant, T the temperature, e the electron charge, I_{dark} the leakage current, and C_{pd} the photodiode capacitance.

These noise sources manifest in distinct ways on the event stream. As visualized in Fig. 3(a), signal events exhibit smooth, motion-aligned trajectories. In contrast, background activity introduces jittery temporal spikes that disrupt coherence, while hot pixels continuously fire at fixed positions, violating spatial consistency. To differentiate such patterns, we implement a state-space classifier that integrates local spatial and temporal neighborhoods:

$$f_\theta : (\mathcal{N}_s(e_i), \mathcal{N}_t(e_i)) \to \{0, 1\}, \tag{4}$$

where $\mathcal{N}_s$ and $\mathcal{N}_t$ denote spatial and temporal neighborhoods. The classifier integrates local geometry and motion continuity to suppress structured spatial and temporal noise.

3.2 EDmamba

Overview Architecture. Fig. 2 illustrates the architecture of **EDmamba**, a dual-branch encoder-decoder framework for event denoising. Raw events are encoded as a spatiotemporal point cloud and structured into a 4D tensor. A Coarse Feature Extraction (CFE) stage applies depthwise convolutions and linear projections to jointly encode geometric and polarity-aware features. To address different noise types, EDmamba includes two decoupled branches: the **Spatial SSM (S-SSM)** targets location-dependent structured noise such as leakage and fixed-pattern effects, while the **Temporal SSM (T-SSM)** handles temporally uncorrelated background activity modeled as Poisson and thermal noise. For directional modeling, the 4D event cloud is flattened into three sequences: two spatial (via space-filling curves) and one temporal (via time scan). These are processed through a U-Net-style hierarchy of multi-scale Spatial-Temporal State Space Blocks (STSSBs), which embed S-SSM and T-SSM modules to model spatial and temporal dependencies. Fused features are decoded to generate denoised events, with skip connections preserving information across scales.

Input Representation. We represent raw events as $E = \{e_i\}_{i=1}^{N}$, where each $e_i = (x_i, y_i, t_i, p_i)$ denotes an event with spatial coordinates (x_i, y_i), timestamp t_i, and polarity $p_i \in \{-1, 1\}$. The event stream is divided into L consecutive segments $\{S_k\}_{k=1}^{L}$, each containing N events. Within each segment S_k, timestamps are normalized

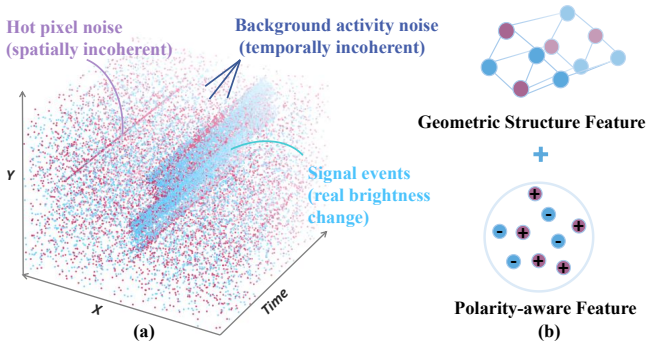

Figure 3: (a) Visualization of signal events and spatiotemporally incoherent noise. (b) Feature decomposition into geometry and polarity components in CFE module. Red and blue dots indicate ON and OFF polarity events, respectively.

using the first and last event times, $t_0 = t^k_{\text{start}}$ and $t_e = t^k_{\text{end}}$, as:

$$z_i = \frac{t_i - t_0}{t_e - t_0}, \quad \text{for } e_i \in S_k, \tag{5}$$

yielding a 3D pseudo point cloud (x_i, y_i, z_i). Incorporating polarity p_i gives a 4D event cloud (x_i, y_i, z_i, p_i) that preserves spatial coordinates, temporal order, and polarity cues. This representation preserves spatial structure, captures local temporal context via normalized timestamps, and retains polarity information in a compact, learnable form.

Event Sampling. To reduce redundancy while retaining salient motion patterns, we apply structured sampling to the high-rate event stream. Events are voxelized by discretizing the normalized timestamp z_i with voxel size v, forming integer coordinates $(x_i, y_i, \lfloor z_i/v \rfloor, p_i)$. A spatial hash function $\mathcal{H} : \mathbb{Z}^4 \to \mathbb{N}$ assigns each voxel a unique key. During training, one event is randomly sampled from each non-empty voxel, preserving statistical diversity while reducing computation and suppressing spurious noise.

Coarse Feature Extraction. To support effective denoising, we propose a Coarse Feature Extraction (CFE) module that decomposes the 4D event cloud into two modality-specific components: geometric structure and polarity signal (Fig. 3(b)). This decomposition reflects two fundamental properties of event data: *(i)* signal events often form coherent geometric patterns aligned with motion; *(ii)* polarity provides an additional modality that is unique to event cameras, and polarity inconsistencies are frequently indicative of noise such as flip errors or unstructured background firing.

Given a segment of events $\mathcal{E}_n = \{(x_i, y_i, z_i, p_i)\}_{i=1}^{M_n}$, CFE applies 1D convolutions with activation functions to extract axis-wise features. The geometric and polarity branches operate on (x, y, z) and (x, y, p) respectively, encoding motion-aligned structures and polarity consistency. Axis-wise convolutions extract modality-specific features, which are subsequently fused via a 1×1 projection:

$$f_i = \text{Conv}_{1\times1}\{(\phi_{\text{geom}}, \phi_{\text{pol}})_{\text{concat}}\}, \tag{6}$$

yielding a compact point-wise embedding f_i that preserves spatial, temporal, and polarity-aware cues. These features are subsequently passed to two decoupled branches, each specialized in suppressing either structured spatial artifacts or stochastic temporal noise.

3.3 S-SSM: Spatial Modeling with Local Geometric Priors.

Spatial noise in event cameras often originates from fixed-pattern leakage, hot pixels, or circuit-level inconsistencies. These recurring artifacts usually exhibit location-dependent repetition and significantly disrupt the surrounding local geometric coherence. To mitigate such structured artifacts, we design a Spatial State-Space Module (S-SSM) that explicitly incorporates spatial priors.

S-SSM leverages space-filling curves (e.g., Z-order, Hilbert) to flatten the 3D spatial domain into sequences while preserving neighborhood continuity. This allows the model to reason over local geometric patterns and edge structures. The sequences are processed by Mamba blocks with depthwise convolutions and bidirectional state updates, capturing both short- and long-range dependencies efficiently. The design biases the model toward spatial smoothness, enabling it to detect structural edges and suppress isolated or repetitive noise. As spatial noise lacks meaningful scene-driven causes, S-SSM focuses on enforcing geometric regularity rather than modeling causality.

3.4 T-SSM: Temporal Modeling with Motion Continuity.

Temporal noise, caused by shot noise, thermal fluctuations, or background activity, severely disrupts the consistency of event streams due to its highly random and unstructured temporal nature. To address this challenge, we introduce the Temporal State-Space Module (T-SSM), which accurately captures motion-consistent patterns by explicitly modeling bidirectional temporal dependencies.

T-SSM first sorts events by normalized timestamp to form a temporally ordered sequence, which is processed by a bidirectional Mamba block to capture forward and backward motion patterns. By learning global temporal consistency, T-SSM suppresses scattered or flickering events while preserving coherent trajectories. This design reflects a key physical prior: real motion yields causally consistent patterns, while temporal noise is fundamentally acausal. To exploit this distinction, T-SSM learns consistent transitions to remove incoherent activations, working alongside the spatial branch for joint spatiotemporal denoising.

4 Experiments

Implementation Details. We optimize EDmamba using a cross-entropy objective, following standard practices in event denoising [9, 10]. The model is implemented in PyTorch and trained for 50 epochs on eight NVIDIA RTX A6000 GPUs with a batch size of 128. We use the AdamW optimizer with an initial learning rate of 8×10^{-5} per sample and a weight decay of 5×10^{-2} to ensure stable convergence and effective regularization. During training, input events are voxelized along the z-axis with a grid resolution of 0.1 and a fixed sample size $N = 10240$, preserving the spatiotemporal structure while maintaining computational efficiency. The network adopts a U-Net-style architecture [26], consisting of a two-stage encoder and a single-stage decoder with block depths of [2, 4] and [2], respectively. The encoder applies serialized pooling (scale factor 2) after the first stage, increasing channels from 8 to 16. The decoder uses serialized unpooling and skip connections for multi-scale fusion.

Methods	DND21 (346 × 260)									
	1 Hz/pixel		3 Hz/pixel		5 Hz/pixel		7 Hz/pixel		10 Hz/pixel	
	Hotel-bar	Driving	Hotel-bar	Driving	Hotel-bar	Driving	Hotel-bar	Driving	Hotel-bar	Driving
EvFlow	0.5137	0.5341	0.5144	0.5351	0.5154	0.5359	0.5160	0.5368	0.5172	0.5379
TS	0.5785	0.6824	0.5721	0.6262	0.5695	0.6250	0.5688	0.6249	0.5689	0.6246
DWF	0.8911	0.6592	0.8616	0.6532	0.8778	0.6502	0.8683	0.6452	0.8563	0.6366
KNoise	0.6780	0.6297	0.6524	0.6203	0.6579	0.6201	0.6489	0.6148	0.6413	0.6146
YNoise	0.7699	0.8086	0.7658	0.8041	0.7594	0.7978	0.7553	0.7949	0.7507	0.7873
RED	0.6475	0.5873	0.6571	0.5913	0.6634	0.5944	0.6721	0.6003	0.6867	0.6062
AEDNet	0.8070	0.8368	0.8568	0.8337	0.9561	0.8325	0.8850	0.8071	0.8990	0.8201
EDnCNN	0.9573	0.8873	0.9371	0.8771	0.9365	0.8748	0.9254	0.8654	0.9006	0.8574
EDformer	0.9928	0.9542	0.9891	0.9472	0.9845	0.9424	0.9793	0.9344	0.9699	0.9264
Ours	**0.9963**	**0.9694**	**0.9949**	**0.9710**	**0.9955**	**0.9734**	**0.9939**	**0.9689**	**0.9916**	**0.9636**

Methods	DVSCLEAN (1280 × 720)									
	Double-bracket		Double-ship		Double-airplane		Multi-helicopter		Multi-car	
	50%	100%	50%	100%	50%	100%	50%	100%	50%	100%
EvFlow	0.6221	0.7591	0.6998	0.6959	0.8100	0.7922	0.8562	0.8368	0.7919	0.7808
TS	0.8303	0.8092	0.8054	0.7919	0.8507	0.8120	0.8749	0.8343	0.8707	0.8438
DWF	0.5995	0.6313	0.5998	0.6325	0.5770	0.5954	0.6201	0.6048	0.6098	0.5957
KNoise	0.5958	0.5765	0.5950	0.5805	0.5751	0.5563	0.5909	0.5713	0.5975	0.5775
YNoise	0.6194	0.6156	0.5903	0.5886	0.6634	0.6529	0.7504	0.7356	0.6540	0.6471
RED	0.5972	0.6792	0.6469	0.6439	0.7357	0.7109	0.7308	0.7062	0.7458	0.7289
AEDNet	0.7314	0.6349	0.7142	0.6450	0.5530	0.5158	0.6509	0.5822	0.6160	0.5409
EDnCNN	0.9432	0.7766	0.9473	0.7776	0.9295	0.7827	**0.9394**	0.7679	**0.9301**	0.7615
EDformer	0.9209	0.8565	0.9382	0.8801	0.8745	0.7924	0.8945	0.8251	0.9056	0.8431
Ours	**0.9457**	**0.9248**	**0.9684**	**0.9572**	**0.9684**	**0.8704**	0.9247	**0.9122**	0.9218	**0.9036**

Table 1: AUC results on DND21 and DVSCLEAN under varying shot noise rates. Best and second best results are highlighted.

Datasets. We evaluate EDmamba on both labeled and unlabeled event datasets across synthetic and real-world domains. Supervised training is conducted on ED24 [10], a large-scale dataset annotated for background activity noise, collected under 21 controlled illumination levels with DAVIS346. Quantitative evaluation is performed on two labeled benchmarks: DVSCLEAN [9], a synthetic dataset generated via ESIM with noise injection, and DND21 [16], a v2e-based dataset with frame-level supervision. To assess generalization and denoising quality, we further evaluate on unlabeled real-world datasets E-MLB [25], which contains diverse indoor and outdoor motions under varying lighting.

Compared Methods. We conduct extensive comparisons with state-of-the-art event denoising methods across both traditional and learning-based categories. For conventional approaches, we evaluate against density-based filters: BAF [13], KNoise [15], DWF [16], and YNoise [27]; time-surface methods: TS [21] and IETS [6]; the recursive event denoiser MLB [25]; optical flow-based method EvFlow [7]; and the guided filter GEF [20]. On the learning-based side, we compare with MLP-based method MLPF [16], CNN-based models EDnCNN [8] and EventZoom [23], the PointNet-based AEDNet [9], and the Transformer-based EDformer [10].

Metrics. For labeled datasets (DVSCLEAN [9] and DND21 [16]), we evaluate denoising performance using the Area Under the Curve (AUC) of event-level predictions, computed from binary ground-truth labels. For the unlabeled dataset E-MLB [25], we adopt the Mean Event Structural Ratio (MESR) [25], which quantifies structural consistency by measuring contrast enhancement in motion-compensated event volumes. Unlike label-dependent metrics, MESR leverages statistical regularities in the event stream and does not require annotations or auxiliary modalities, making it suitable for real-world evaluation.

4.1 Quantitative Evaluation

To evaluate denoising performance on labeled datasets, we compute AUC scores following the evaluation protocol in [16], with results reported in Tab. 1. DND21 includes two 346×260 test scenes with shot noise rates from 1–10 Hz/pixel, simulating low-light conditions. DVSCLEAN provides five 1280×720 sequences under two noise levels (50% and 100%). Our EDmamba achieves the highest AUC scores on DND21 by effectively handling varying shot noise levels. On average, EDmamba achieves AUC scores of 0.9944 and 0.9693 on the hotel-bar and driving scenes, outperforming EDformer with relative AUC improvements of 1.15% and 3.01%. While EDmamba achieves leading performance on most DVSCLEAN sequences, EDnCNN slightly outperforms it in a few cases due to its use of k-nearest spatio-temporal neighbors for fine-grained event aggregation. However, this explicit neighbor search increases computational cost (Tab. 3), whereas EDmamba processes raw streams directly with higher efficiency.

To further evaluate the generalization of EDmamba in label-free, real-world scenarios, we conduct MESR testing on the E-MLB (Daylight, Night) and DND21 datasets. As shown in Tab. 2, EDmamba consistently delivers strong MESR performance, demonstrating robust denoising under challenging lighting. While methods like TS tend to achieve high MESR by over-suppressing both noise and informative background content, EDmamba strikes a better balance between denoising and content preservation.

4.2 Qualitative Evaluation

To illustrate the effectiveness of our method, we present visual comparisons across different datasets and noise levels. Fig. 4 presents qualitative comparisons on the E-MLB dataset under daytime and

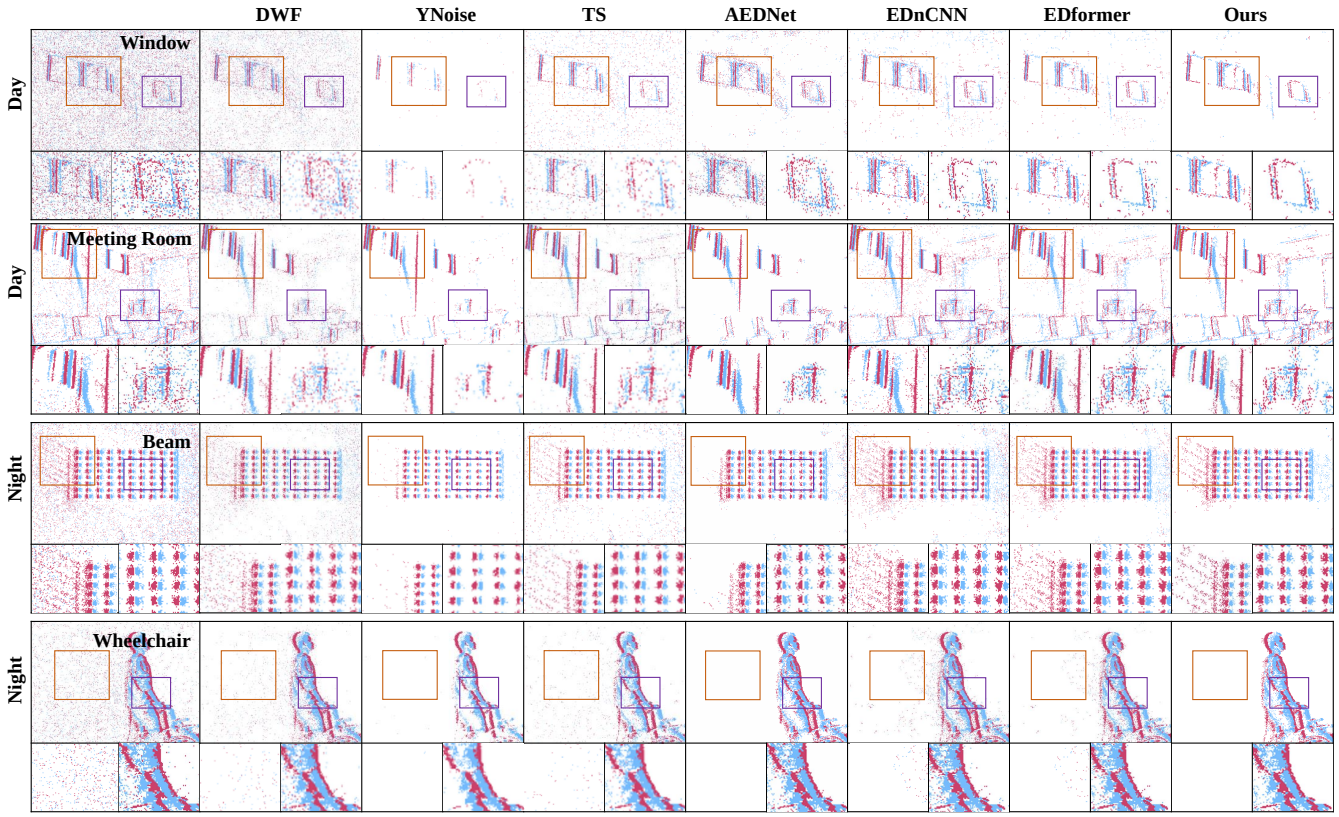

Figure 4: Visual comparison on the E-MLB [25] dataset under daytime (top two rows) and nighttime (bottom two rows) conditions. EDmamba effectively suppresses background noise while preserving fine motion and structural details. In contrast, baseline methods either leave residual noise or blur object contours, especially under low-light conditions.

Method	E-MLB (Daylight)				E-MLB (Night)				DND21
	ND1	ND4	ND16	ND64	ND1	ND4	ND16	ND64	-
Raw	0.821	0.824	0.815	0.786	0.890	0.824	0.786	0.768	0.869
TS [21]	0.943	0.955	0.980	**0.995**	0.938	0.884	0.859	0.907	0.954
EvFlow [7]	0.871	0.919	0.917	0.921	0.951	0.876	0.852	0.886	1.034
IETS † [6]	0.772	0.785	0.777	0.753	0.950	0.823	0.804	0.711	0.900
KNoise [15]	0.787	0.819	0.810	0.786	0.904	0.842	0.819	0.860	0.998
EDnCNN [8]	0.887	0.908	0.903	0.912	1.001	1.024	1.079	1.086	0.977
YNoise [27]	0.902	0.922	0.917	0.934	0.962	0.895	0.874	0.928	0.984
GET † [20]	**1.051**	0.938	0.935	0.927	1.027	0.955	0.946	0.935	0.932
EventZoom † [23]	0.996	0.988	**0.996**	0.970	**1.055**	1.007	1.010	0.988	**1.059**
MLPF [16]	0.851	0.855	0.846	0.840	0.926	0.928	0.910	0.906	0.944
DWF [16]	0.932	0.945	0.943	0.904	0.916	0.871	0.825	0.873	0.972
AEDNet [9]	0.789	0.836	0.803	0.789	0.887	0.929	0.929	0.958	0.919
RED [25]	0.971	0.943	0.946	0.923	0.948	0.973	1.001	0.916	0.945
EDformer [10]	0.952	0.955	0.956	0.942	1.048	1.019	1.076	**1.099**	1.041
Ours	0.976	**0.990**	0.985	0.972	1.002	**1.025**	**1.082**	1.089	1.057

†: The result is derived from E-MLB [25], as the official code is not publicly available.

Table 2: The MESR results of denoising methods on E-MLB and DND21 datasets. Best and second-best values are highlighted.

nighttime conditions. EDmamba preserves structural details across diverse scenes. In the *Window* and *Meeting Room* examples, it retains fine architectural contours such as frame lines and corners, while suppressing background clutter. In contrast, other methods (e.g., DWF, YNoise) tend to oversmooth or break weak edges.

In nighttime *Beam* scenes, EDmamba demonstrates robustness by preserving low-intensity, elongated light trails that are otherwise

Method	GFLOPs	#Params	Inf. Time (s)	Rel. Speed
TS	N/A	N/A	0.1296	1.0×
EvFlow	N/A	N/A	1.5545	0.08×
DWF	N/A	N/A	0.0954	1.36×
KNoise	N/A	N/A	**0.0198**	**6.55**×
YNoise	N/A	N/A	0.0513	2.53×
RED	N/A	N/A	2.2716	0.06×
EDnCNN	234.51	614.55K	20.1885	1.0×
AEDNet	4400.46	45.87M	43.4250	0.46×
EDformer	8.41	**49.80K**	2.4943	8.09×
Pre-Mamba†	6.23	264.63K	0.0987	204.54×
Joint-SSM	5.17	102.91K	0.0931	216.85×
Ours	**2.27**	88.98K	**0.0685**	**294.72**×

†: The result is derived from Pre-Mamba [12].

Table 3: Efficiency comparison in terms of GFLOPs, parameters, inference time, and relative speed (100K events).

removed by YNoise and AEDNet. This highlights our model's ability to distinguish faint but coherent motion patterns from stochastic noise. These results validate the model's capacity for structure-aware denoising across lighting conditions. Due to the page limit, additional qualitative results on two DND21 test scenes under two noise levels are provided in the supplementary material.

4.3 Model Complexity and Efficiency Comparison

We compare the computational efficiency of all methods in Tab. 3, including FLOPs, model size, and inference time on 100K events (NVIDIA RTX A6000). Filtering-based methods (e.g., KNoise, YNoise) are fast due to their lightweight, non-learnable design but generalize poorly. In contrast, learning-based models like EDnCNN and AEDNet are slower, with larger models and inference times over 20s.

EDformer reduces model size using a lightweight Transformer backbone, but its inference is limited by the quadratic complexity of self-attention and joint modeling, taking 2.49 seconds for 100K events. While Pre-Mamba was originally proposed for deraining, it adopts a joint 4D state-space model that entangles spatial and temporal dynamics, resulting in higher computational cost and 3× more parameters than our decoupled design. To further validate the benefit of decoupled spatial-temporal modeling, we design a control variant named Joint-SSM as a comparative baseline. Instead of using two separate Mamba branches, we adopt a shared Mamba block to jointly encode spatial and temporal sequences. The spatial and temporal event streams are first flattened via scan operations and concatenated before being fed into the shared Mamba. Due to the lack of task-specific specialization, this design requires more parameters and slower inference to achieve comparable performance, underscoring the necessity of noise-specific modeling.

Our method achieves a superior trade-off between speed and capacity. With 88.98K parameters and 2.27 GFLOPs, it processes 100K events in 0.0685 seconds, which is 36× faster than EDformer and 1.4× faster than Pre-Mamba. These results highlight the effectiveness of our design in balancing efficiency and real-time applicability.

Method Variant	AUC (%)		Δ	
	Hotel-bar	Driving	Hotel-bar	Driving
Full Model	**99.55**	**97.34**	–	–
w/o Geometry Feat.	99.24	97.12	-0.31	-0.22
w/o Polarity Feat.	99.19	96.10	-0.36	-1.24
w/o S-SSM	98.78	95.70	-0.77	-1.64
w/o T-SSM	98.69	94.66	-0.86	-2.68
Joint-SSM	99.20	96.49	-0.35	-0.85

Table 4: Ablation study on DND21 at 5 Hz/pixel. We report AUC scores (%) for two scenes (Hotel-bar and Driving), along with performance drops (Δ) from the full model.

4.4 Ablation Experiments

To assess the role of each component, we conduct ablation studies on the DND21 dataset (Tab. 4). Disabling either input feature causes performance drops, with polarity having a greater impact, as it encodes signal activity and reveals polarity-related noise unique to event data. Removing either S-SSM or T-SSM causes a larger degradation, indicating that temporal and spatial modeling plays a more critical role than feature encoding alone. Among the two branches, T-SSM contributes more in motion-heavy scenarios such as Driving, highlighting the effectiveness of temporally ordered modeling for motion continuity. We also compare against the Joint-SSM variant introduced above. Despite using more parameters and slower inference, it still underperforms our decoupled design, reinforcing that task-specific modeling is more effective than simply increasing model size.

5 Conclusion

This paper presents EDmamba, an efficient event denoising framework built on decoupled spatial and temporal state space modeling. By explicitly modeling distinct noise patterns through two specialized Mamba branches, EDmamba achieves high denoising accuracy with reduced model size and inference latency. Our Coarse Feature Extraction module captures both polarity information and geometric structure, while the decoupled design ensures efficient and task-specific processing. Extensive experiments across synthetic and real-world datasets show that EDmamba outperforms prior methods in both accuracy and efficiency. Our design highlights the value of decoupled spatiotemporal modeling for event noise, offering a new perspective on architectural specialization in event-based learning. Future work will explore integrating EDmamba into downstream systems and deployment on resource-constrained platforms, pushing forward practical and scalable event-based perception.

References

[1] Rui Jiang, Xiaozheng Mou, Shunshun Shi, Yueyin Zhou, Qinyi Wang, Meng Dong, and Shoushun Chen. 2020. Object tracking on event cameras with offline–online learning. *CAAI Transactions on Intelligence Technology* 5, 3 (2020), 165–171.

[2] Daniel Gehrig and Davide Scaramuzza. 2024. Low-latency automotive vision with event cameras. *Nature* 629, 8014 (2024), 1034–1040.

[3] Davide Falanga, Kevin Kleber, and Davide Scaramuzza. 2020. Dynamic obstacle avoidance for quadrotors with event cameras. *Science Robotics* 5, 40 (2020), eaaz9712.

[4] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. 2020. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence* 44, 1 (2020), 154–180.

[5] Yang Feng, Hengyi Lv, Hailong Liu, Yisa Zhang, Yuyao Xiao, and Chengshan Han. 2020. Event density based denoising method for dynamic vision sensor.

Applied Sciences 10, 6 (2020), 2024.

[6] R Wes Baldwin, Mohammed Almatrafi, Jason R Kaufman, Vijayan Asari, and Keigo Hirakawa. 2019. Inceptive event time-surfaces for object classification using neuromorphic cameras. In *Image Analysis and Recognition: 16th International Conference, ICIAR 2019, Waterloo, ON, Canada, August 27–29, 2019, Proceedings, Part II 16*. Springer, 395–403.

[7] Yanxiang Wang, Bowen Du, Yiran Shen, Kai Wu, Guangrong Zhao, Jianguo Sun, and Hongkai Wen. 2019. EV-gait: Event-based robust gait recognition using dynamic vision sensors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6358–6367.

[8] R Baldwin, Mohammed Almatrafi, Vijayan Asari, and Keigo Hirakawa. 2020. Event probability mask (epm) and event denoising convolutional neural network (edncnn) for neuromorphic cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1701–1710.

[9] Huachen Fang, Jinjian Wu, Leida Li, Junhui Hou, Weisheng Dong, and Guangming Shi. 2022. AEDNet: Asynchronous event denoising with Spatial-Temporal correlation among irregular data. In *Proceedings of the 30th ACM International Conference on Multimedia*. 1427–1435.

[10] Bin Jiang, Bo Xiong, Bohan Qu, M Salman Asif, You Zhou, and Zhan Ma. 2024. EDformer: Transformer-Based Event Denoising Across Varied Noise Levels. In *European Conference on Computer Vision*. Springer, 200–216.

[11] Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752* (2023).

[12] Ciyu Ruan, Ruishan Guo, Zihang Gong, Jingao Xu, Wenhan Yang, and Xinlei Chen. 2025. PRE-Mamba: A 4D State Space Model for Ultra-High-Frequent Event Camera Deraining. *arXiv preprint arXiv:2505.05307* (2025).

[13] Tobi Delbruck et al. 2008. Frame-free dynamic digital vision. In *Proceedings of Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society*, Vol. 1. Citeseer, 21–26.

[14] Hongjie Liu, Christian Brandli, Chenghan Li, Shih-Chii Liu, and Tobi Delbruck. 2015. Design of a spatiotemporal correlation filter for event-based sensors. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 722–725.

[15] Alireza Khodamoradi and Ryan Kastner. 2018. $O(N)$ O (N)-space spatiotemporal filter for reducing noise in neuromorphic vision sensors. *IEEE Transactions on Emerging Topics in Computing* 9, 1 (2018), 15–23.

[16] Shasha Guo and Tobi Delbruck. 2022. Low cost and latency event camera background activity denoising. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 1 (2022), 785–795.

[17] Sio-Hoi Ieng, Christoph Posch, and Ryad Benosman. 2014. Asynchronous neuromorphic event-driven image filtering. *Proc. IEEE* 102, 10 (2014), 1485–1499.

[18] Daniel Czech and Garrick Orchard. 2016. Evaluating noise filtering for event-based asynchronous change detection image sensors. In *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 19–24.

[19] Jinjian Wu, Chuanwei Ma, Xiaojie Yu, and Guangming Shi. 2020. Denoising of event-based sensors with spatial-temporal correlation. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 4437–4441.

[20] Peiqi Duan, Zihao W Wang, Boxin Shi, Oliver Cossairt, Tiejun Huang, and Aggelos K Katsaggelos. 2021. Guided event filtering: Synergy between intensity images and neuromorphic events for high performance imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 11 (2021), 8261–8275.

[21] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E Shi, and Ryad B Benosman. 2016. Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions on pattern analysis and machine intelligence* 39, 7 (2016), 1346–1359.

[22] Xuemei Xie, Jiang Du, Guangming Shi, Jianxiu Yang, Wan Liu, and Wang Li. 2018. DVS image noise removal using K-SVD method. In *Ninth International Conference on Graphic and Image Processing (ICGIP 2017)*, Vol. 10615. SPIE, 1099–1107.

[23] Peiqi Duan, Zihao W Wang, Xinyu Zhou, Yi Ma, and Boxin Shi. 2021. EventZoom: Learning to denoise and super resolve neuromorphic events. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12824–12833.

[24] Yusra Alkendi, Rana Azzam, Abdulla Ayyad, Sajid Javed, Lakmal Seneviratne, and Yahya Zweiri. 2022. Neuromorphic camera denoising using graph neural network-driven transformers. *IEEE Transactions on Neural Networks and Learning Systems* 35, 3 (2022), 4110–4124.

[25] Saizhe Ding, Jinze Chen, Yang Wang, Yu Kang, Weiguo Song, Jie Cheng, and Yang Cao. 2023. E-MLB: Multilevel benchmark for event-based camera denoising. *IEEE Transactions on Multimedia* 26 (2023), 65–76.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[27] Yang Feng, Hengyi Lv, Hailong Liu, Yisa Zhang, Yuyao Xiao, and Chengshan Han. 2020. Event Density Based Denoising Method for Dynamic Vision Sensor. *Applied Sciences* (2020).