S3Net: Spatiotemporally Separated Sparse Network for Neuromorphic Vision Processing

Ping He[1], Rong Xiao[1]*, Wanying Xu[1], Chenwei Tang[1], Shudong Huang[1], Huajin Tang[2]

[1]College of Computer Science, Sichuan University and Engineering Research Center of Machine Learning and Industry Intelligence, Ministry of Education, Chengdu, China
hping.scu@gmail.com, xwanying@stu.scu.edu.cn, rxiao@scu.edu.cn, tangchenwei@scu.edu.cn, huangsd@scu.edu.cn
[2]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
htang@zju.edu.cn

Abstract

Dynamic Vision Sensor (DVS) asynchronously records sparse events triggered by changes in pixel intensity, offering high temporal resolution and low latency. Existing frame-based methods process event data densely, violating its inherent sparsity and introducing computational redundancy. While asynchronous models preserve the event stream's native format, they often neglect spatial information, compromising their adaptability and efficiency. To address these limitations, we propose a Spatiotemporally Separated Sparse Network (S3Net) for efficient event stream encoding and learning. Specifically, we employ a learnable sparse encoding scheme to construct a voxel-structured representation that effectively extracts spatiotemporal relationships among event data. After that, we propose a dual-branch architecture to capture localized spatial dependencies and dynamic temporal patterns of event data. By explicitly decoupling spatial and temporal modeling, S3Net enables end-to-end asynchronous processing of variable-length event sequences, achieving both strong representational capacity and high computational efficiency. Experimental results on six event-based datasets demonstrate that S3Net achieves state-of-the-art performance. Compared to frame-based methods, it significantly reduces computational overhead and model complexity, while also outperforming existing asynchronous approaches in inference speed without compromising accuracy. Extensive experiments across six event-based datasets show that S3Net establishes new state-of-the-art performance. Our method reduces computational costs by 35% and model parameters by 27% compared to frame-based approaches, while delivering 1.58× faster inference than existing point-based methods at comparable accuracy levels.

Code — https://github.com/hppeace/S3Net

Introduction

Dynamic Vision Sensor (DVS) output sparse streams of events only when luminance changes occur, resulting in extremely low latency and high dynamic range (AliAkbarpour et al. 2024; Gallego et al. 2020; Qin et al. 2025). This asynchronous and sparse sensing paradigm enables the precise capture of subtle motion patterns and rapid scene dynamics, making DVS particularly well-suited for high-speed and resource-constrained vision applications. Efficient processing of such event data allows for the construction of informative spatiotemporal representations that facilitate salient feature extraction, suppress noise, and enhance the expressive power of learned models (Feng et al. 2020; Khan, Iqbal, and Martini 2020).

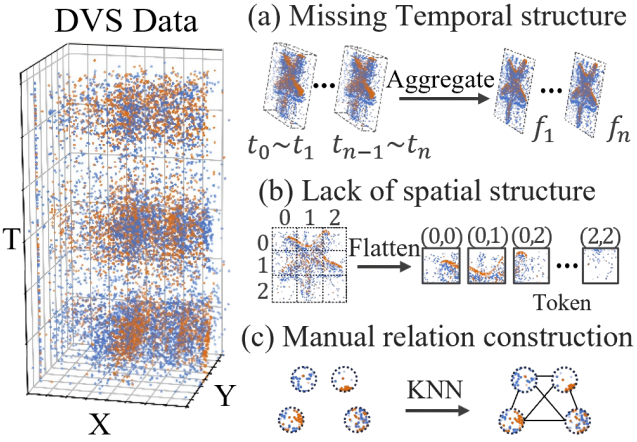

Figure 1: Encoding methods for DVS data, categorized into frame (a) and point representations. The point representation includes token (b) and graph (c) approaches.

Inspired by the success of deep learning in conventional computer vision (He et al. 2016; Han et al. 2022; Liu, Zhang, and Zhang 2024), early work in event-based vision largely relied on handcrafted features and frame-based representations. Rebecq et al. (Benosman et al. 2013) proposed accumulating events over fixed time windows to generate frame-like representations, but this discards precise temporal information. To better model local spatiotemporal patterns, Lagorce et al. (Lagorce et al. 2016) introduced HOTS, which encodes recent temporal activity in an event's spatial neighborhood. However, its reliance on short-term history makes it sensitive to noise. Sironi et al. (Sironi et al. 2018) improved robustness by proposing HATS, which aggregates events within a spatiotemporal window using a local

*Corresponding author: Rong Xiao.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

memory mechanism. Similarly, Gehrig et al. (Gehrig et al. 2019) proposed the Event Spike Tensor (EST), a grid representation that uses bilinear interpolation to project events onto their nearest spatiotemporal grid cells, enabling end-to-end learning via differentiable convolutional networks. But handcrafted features are limited in representational power and generalization, while frame-based processing ignores the inherent asynchrony and sparsity of event data, leading to redundant computation and underutilization of neuromorphic sensing advantages.

Recent advances in event-based vision have led to the development of trainable feature extraction methods that leverage the asynchronous and sparse nature of event streams, as shown in Figure 1. One approach models event data as structured temporal sequences to retain its inherent temporal granularity and continuity. For instance, EventMamba (Ge et al. 2025) introduces random spatial windowing and Hilbert curve scanning to enhance translation equivariance and spatiotemporal locality. Similarly, SMamba (Yang et al. 2025) proposes spatiotemporal continuity assessment and information-prioritized local scanning to sparsify token sequences, thereby improving efficiency in event-based object detection. Alternatively, events are often conceptualized as sparse spatiotemporal samples, prompting the design of architectures that operate directly in event space. Sekikawa et al. (Sekikawa, Hara, and Saito 2019) and Carmen et al. (Martin-Turrero et al. 2024) leveraged PointNet-inspired (Qi et al. 2017), max-pooling to extract local features from raw asynchronous events, enabling the construction of synchronized feature representations.

In addition, Bi et al. (Bi et al. 2019, 2020) were among the first to introduce graph neural networks (GNNs) (Abadal et al. 2021) into event-based vision, treating individual events as graph nodes and constructing spatiotemporal graphs based on neighborhood radii. They used graph convolution to model spatial and temporal relationships among events. Building upon this idea, Li et al. (Li et al. 2021) and Schaefer et al. (Schaefer, Gehrig, and Davide 2022) introduced sliding window mechanisms to enable local updates within asynchronous GNN architectures. To further enhance the perception of local structure, Xie et al. (Xie et al. 2022) proposed Multi-View Voxel Convolution method, which aggregates multiple spatiotemporally adjacent events into voxel blocks to improve the discriminative power of vertex representations. Existing methods still struggle to effectively balance asynchronous processing, sparse representation, and model performance. Event streams exhibit strong temporal continuity and complex spatial structures, yet most current approaches either focus on a single aspect or rely on manually constructed spatiotemporal relationships, limiting flexibility and adaptability. Therefore, designing a unified processing framework that efficiently and robustly captures the spatiotemporal characteristics of event data remains a critical challenge.

Inspired by the spatiotemporal separation strategies commonly employed in video data processing (Bertasius, Wang, and Torresani 2021), we propose a Spatiotemporally Separated Sparse Network (S3Net) for asynchronous event stream learning. The proposed framework explicitly decouples spatial and temporal modeling, enabling effective integration of localized spatial feature extraction and temporal pattern aggregation along the event sequence for multi-scale representation learning. S3Net establishes an asynchronous, sparsity-driven, and end-to-end trainable processing pipeline that significantly improves inference efficiency while maintaining strong representational capacity and accuracy. The main contributions of this work are summarized as follows:

1) We propose a learnable voxel-based sparse encoding method integrated into an end-to-end event stream framework, preserving the inherent sparsity and temporal characteristics of event data while enabling efficient and expressive feature representation.
2) We propose a spatiotemporally separated feature extraction network that supports asynchronous processing of variable-length event streams, enabling adaptive feature encoding for diverse temporal dynamics.
3) The proposed framework achieves state-of-the-art performance on several representative neuromorphic vision datasets with reduced computational overhead and model size, enabling fast and efficient inference.

Preliminary

This section introduces the fundamental concepts underlying the proposed S3Net. Specifically, we briefly overview DVS, sparse convolution, and Mamba.

Dynamic Vision Sensor

DVS asynchronously records changes in pixel-level brightness. Each pixel independently emits an event when the logarithmic intensity change exceeds a predefined threshold, resulting in a sparse event represented as tuples (x, y, t, p), where (x, y) denotes the pixel location, $t \in (0, \infty)$ is the timestamp, and $p \in \{-1, +1\}$ indicates the polarity of intensity change. The event stream ε is formally defined as:

$$\varepsilon = \{e^i\}_{i=1}^{N} = \{(x^i, y^i, t^i, p^i)\}_{i=1}^{N} \tag{1}$$

Unlike conventional image sensors that output redundant frames at fixed intervals, DVS captures only salient changes in the scene, producing asynchronous and sparse event streams. Despite its advantages, the resulting data exhibits complex spatiotemporal structure and is highly susceptible to sensor noise, posing significant challenges for traditional vision algorithms in modeling spatiotemporal dependencies and maintaining robustness.

Sparse Convolution

Sparse convolution (Yan, Mao, and Li 2018) is a variant of traditional convolution designed to efficiently process data with sparse spatial distributions, such as 3D point clouds. Unlike standard convolution operations that perform dense computation over the entire grid, sparse convolution restricts computation to the set of active (non-zero) locations $\mathcal{A}_{in}$, significantly reducing both computational cost and memory usage. Given a sparse input feature map $\mathbf{I} \in \mathbb{R}^{M \times C_{in}}$, where M denotes the number of active sites, and a convolutional

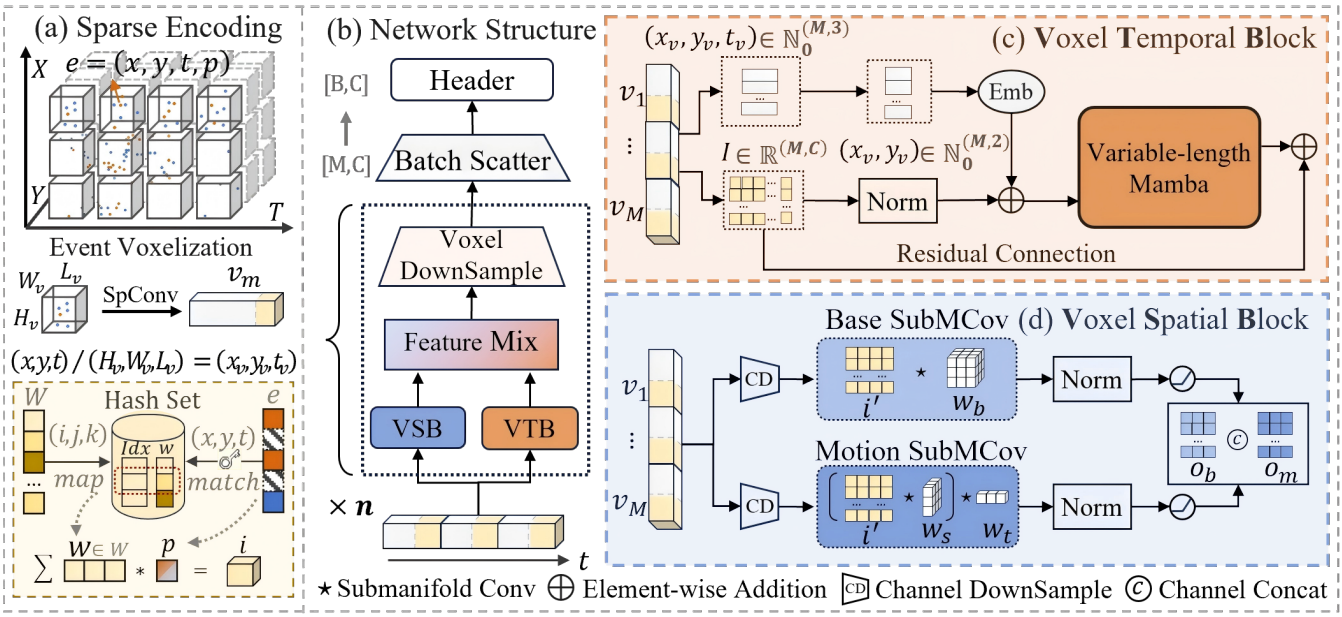

Figure 2: The architecture of S3Net includes sparse encoding (a) and a spatiotemporal feature extractor (b), which contains the Voxel Spatial Block (c) for spatial modeling and the Voxel Temporal Block (d) for temporal modeling.

kernel $\mathbf{W} \in \mathbb{R}^{K \times C_{in} \times C_{out}}$ with kernel size K, the sparse convolution at an active location i is computed as:

$$\mathbf{O}(i) = \sum_{\delta \in K} \mathbf{W}(\delta) \cdot \mathbf{I}(i + \delta), \quad i \in \mathcal{A}_{out}, \tag{2}$$

where $\mathcal{A}_{out}$ denotes the set of active output locations. Computations are performed exclusively over active input locations $(i+\delta)$, enabled by efficient hash-based indexing structures. An important variant of this operation is submanifold sparse convolution (Graham and van der Maaten 2017), which enforces $\mathcal{A}_{\text{out}} = \mathcal{A}_{\text{in}}$. This constraint effectively prevents the spread of activations into empty regions and maintains the sparsity pattern throughout the network. Sparse convolution thus serves as an efficient solution for preserving spatial locality, reducing redundancy, and enabling hierarchical feature extraction in sparse data domains.

Mamba

Mamba (Gu and Dao 2023; Dao and Gu 2024), the temporal modeling backbone adopted in S3Net, is built upon the State Space Model (SSM) (Hamilton 1994), a mathematical framework that describes temporal dependencies through latent state dynamics. Compared with traditional recurrent or attention-based models, SSM-based architectures provide a more efficient and scalable solution for long-range sequence modeling. Formally, a linear time-invariant (LTI) SSM is defined as:

$$\begin{aligned} s'_t &= As_{t-1} + Bi_t \\ o_t &= Cs_t + Di_t, \end{aligned} \tag{3}$$

where s_t is the latent state, i_t is the input, o_t is the output, and matrices A, B, C, and D determine the state transitions and input-output mappings. By exploiting the efficient linear recurrence and continuous temporal modeling of SSM, Mamba enables efficient modeling of long-range dependencies with linear complexity. In the context of event-based processing, such a structure is particularly beneficial for capturing fine-grained temporal dynamics in irregular and asynchronous input streams.

Method

In this section, we begin by presenting an overview of the overarching framework of S3Net. Subsequently, we delve into detailed explanations of its two components: sparse encoding and network structure.

Overview

We propose a framework named S3Net, specifically designed for event stream processing, as illustrated in Figure 2. The processing pipeline begins with the sparse coding module, which aligns raw asynchronous events onto a three-dimensional spatiotemporal grid and encodes them into sparse voxel representations with a fixed size (H_v, W_v, L_v), preserving both the inherent sparsity and temporal ordering of the input. The voxelized features are then passed through a series of stacked feature extraction blocks. Each block consists of two dedicated branches: the voxel spatial block and the voxel temporal block, which independently capture spatial structure and temporal dynamics, respectively. Their outputs are fused to form a unified and expressive representation. To gradually reduce spatial resolution and construct hierarchical feature representations, voxel downsampling layers are inserted between every two feature extraction blocks. Each downsampling layer includes a sparse convolution, batch normalization, and an activation function,

enabling efficient spatial reduction while maintaining discriminative capacity.

Since the number of non-empty voxels varies across samples, we apply a batch-wise scatter aggregation to produce a fixed-length global feature for each sample, enabling consistent input to the downstream classification head regardless of voxel count. The aggregation is defined as:

$$\mathbf{O}_b = \frac{1}{N_b} \sum_{m:m\in b} \mathbf{I}_m, \quad \text{for } b = 0, 1, \ldots, B-1, \tag{4}$$

where $\mathbf{I}_m$ denotes the feature of the m-th voxel, and N_b is the number of voxels belonging to the b-th sample in the batch. The resulting representations $\mathbf{O}_b$ are then passed to a lightweight classification head for downstream tasks.

Sparse Encoding

Although event streams are inherently continuous and asynchronous in the temporal domain, their highly irregular distribution and variable sequence lengths across samples pose significant challenges for direct processing with neural networks, such as inconsistent input sizes and redundant information. To preserve temporal structure while improving computational efficiency, we adopt a voxel-based encoding strategy that transforms raw event streams into sparse and structured spatiotemporal representations.

Specifically, we first define a fixed temporal resolution t_w to discretize the time axis. Each event, based on its timestamp t, is then mapped to a corresponding discrete time. This process aligns asynchronous events to a coarser but regular temporal resolution, making them more suitable for downstream neural architectures.

$$t = \lfloor (t - t_{min}) / t_w \rfloor \tag{5}$$

Due to this downsampling, multiple events may fall into the same spatiotemporal point. To further simplify the representation and avoid redundancy, we retain only one representative event per polarity channel within each point. Specifically, we construct a binary feature vector $\mathbf{I} = [f^-, f^+]$, where $f^- = 1$ indicates the presence of at least one negative-polarity event, and $f^+ = 1$ indicates the presence of a positive-polarity event.

$$\mathbf{I} = [f^-, f^+], f^p = \mathbb{1}\left(\exists\,(x_j, y_j, t_j) \text{ s.t. } p_j = p\right) \tag{6}$$

Here, $\mathbb{1}(\cdot)$ denotes the indicator function, which evaluates to 1 if the condition inside holds true, and 0 otherwise. After temporal downsampling, the event stream is processed by two layers of 3D sparse convolution to construct an initial representation. The encoding module outputs two components. The first is a voxel feature tensor $\mathbf{I} \in \mathbb{R}^{M\times 2}$, where the two channels correspond to positive and negative polarities. The second is a coordinate tensor $(x_v, y_v, t_v) \in \mathbb{N}_0^{M\times 3}$.

$$(x_v, y_v, t_v) = (x, y, t)/(H_v, W_v, L_v) \tag{7}$$

The M non-empty voxels are sorted in ascending order according to the earliest event timestamp within each voxel, rather than arranged arbitrarily. This temporal ordering ensures a consistent and semantically meaningful input sequence for downstream models that rely on temporal structure. The resulting representation achieves simultaneous sparsity and temporal alignment, making it particularly suitable for sparse convolutional backbones and sequence-aware temporal modules.

Network Structure

Given that the encoded event data varies in length across samples, we aim to decouple its spatiotemporal features while maintaining asynchronous and sparse processing. To this end, we design a dedicated feature extraction block consisting of two modules. Voxel Temporal Block (VTB) captures temporal dynamics, while Voxel Spatial Block (VSB) focuses on spatial structure. These modules extract complementary features along different dimensions, and their outputs are fused via channel-wise addition for efficient integration of spatiotemporal information.

VTB. To efficiently model long event sequences, we adopt the Mamba architecture within the VTB. Unlike Transformers, which rely on attention mechanisms, Mamba utilizes linear state-space models to achieve significantly higher computational efficiency. This makes it particularly well-suited for processing DVS event data characterized by long temporal spans. However, the original Mamba does not support variable-length sequences and lacks mechanisms to prevent information leakage when processing packed batches. Directly applying it in such settings may lead to cross-sample interference and degraded modeling accuracy.

To overcome this limitation, we adopt the position-index-based masking strategy and segment-wise computation mechanism proposed in prior work (Xu et al. 2024). Specifically, we incorporate their modifications to Mamba's internal state-space model and convolutional operators, enabling efficient packed-sequence processing without inter-sample information leakage. This design eliminates the need for traditional padding and ensures accurate and efficient temporal modeling across variable-length inputs.

Moreover, since Mamba is inherently a linear state-space model with limited sensitivity to spatial structure, we explicitly embed the spatial coordinates (x_v, y_v) into each voxel feature as positional encodings. This enhances the model's capacity to capture spatial relationships within event sequences. In addition, residual connections and layer normalization are integrated into the network architecture to improve training stability and strengthen representational expressiveness.

$$\mathbf{O} = \mathbf{I} + \text{Mamba}(\text{Emb}(x_v, y_v) + \mathbf{I}) \tag{8}$$

VSB. VSB is a sparse voxel processing module designed to explicitly decouple the modeling of spatial structures and temporal dynamics. Although voxelized event data is represented as a 3D tensor, it essentially originates from a sequence of 2D images evolving over time. As a result, spatial and temporal patterns exhibit distinct characteristics. Directly modeling them without separation often leads to feature entanglement and degraded representation. To address

Methods	Async.	Input	Network	GFlops	Params (M)	Time (ms)	Top-1 acc. (%)
EST	✗	F	CNN	4.28	21.38	10.8	<u>81.7</u>
MVF-Net	✗	F	CNN	5.62	33.62	14.3	64.2
Matrix-LSTM	✗	F	CNN	4.82	21.43	16.6	73.8
AsyNet	✓	F	CNN	-	3.70	-	74.5
EV-VGCN	✗	P	GNN	0.70	0.84	14.8	74.8
VMV-GCN	✗	P	GNN	1.30	0.86	11.6	77.8
EDGCN	✗	P	GNN	0.57	0.77	-	80.1
EVSTr	✗	P	GNN	0.34	0.93	13.3	79.7
AEGNN	✓	P	GNN	-	20.4	-	66.8
SlideGCN	✓	P	GNN	-	-	-	76.1
S3Net (this work)	✓	P	CNN+Mamba	1.52	5.88	7.3	**83.5**

Table 1: Comparison of representative methods by asynchronous processing, input type, network, computation (GFlops), parameters, inference time, and top-1 accuracy. **F** and **P** denote frame- and point-based input, respectively. Best results are in bold, second-best are underlined, and $-$ indicates not reported. For methods without official implementations, we report runtime referenced from (Liu, Wang, and Sun 2023), and evaluate our method under similar hardware settings for fair comparison.

this, we design the structurally decoupled VSB module, which extracts spatial and motion features independently through separate branches, followed by effective fusion to enable complementary information integration.

This module integrates a channel mapping layer, dual-branch submanifold convolutional paths, and a feature fusion unit. Given a sparse voxel feature tensor $\mathbf{I} \in \mathbb{R}^{M \times C}$, two independent channel downsampling layers first generate inputs for the Base and Motion branches, each with $\frac{1}{2}C$ channels, enabling separate modeling of spatial structures and temporal-motion dynamics. The Base branch adopts a standard submanifold convolution with kernel $W^b \in \mathbb{R}^{3\times3\times3}$ to capture spatial structures within full 3D neighborhoods. In contrast, inspired by the concept of depthwise–pointwise convolution, the Motion branch first applies a convolution $W_s \in \mathbb{R}^{3\times3\times1}$ to extract intra-frame spatial features, followed by a convolution $W_t \in \mathbb{R}^{1\times1\times3}$ to capture inter-frame temporal dynamics. This design reduces channel redundancy and minimizes interference between spatial and temporal domains. Finally, the outputs of the two branches are concatenated along the channel dimension to form a unified representation that integrates both spatial structures and motion patterns, providing more discriminative voxel features for downstream processing.

Experiments

We evaluate the proposed S3Net on six event-based classification datasets. This section first introduces the experimental settings, followed by performance comparisons with existing methods and ablation studies to validate the effectiveness of our framework.

Experimental Setup

Datasets. We evaluate our method on six representative event-based classification datasets. N-Caltech101 (Orchard et al. 2015) and CIFAR10-DVS (Li et al. 2017) are converted from frame-based image datasets using saccadic motion, containing 101 and 10 object categories, respectively. N-CARS (Sironi et al. 2018) comprises real-world driving scenes for binary classification (car vs. background). N-MNIST (Orchard et al. 2015) is recorded from MNIST digits with fixed saccades. DVSGesture (Amir et al. 2017) and ASL-DVS (Bi et al. 2020) focus on gesture recognition, with the former covering 11 dynamic gestures and the latter including over 100,000 samples of static American Sign Language letters.

Implementation Details. For dataset preparation, we follow the original train-test splits for N-CARS, N-MNIST, and ASL-DVS. For CIFAR10-DVS, N-Caltech101, and DVS-Gesture, we randomly split the samples such that 90% are used for training and 10% for testing. To enhance generalization, we apply the same data augmentation pipeline across all datasets without any dataset-specific tuning. The augmentations include FlipEvents, ShearEvents, TranslateEvents, CropEvents, and DropoutEvents, all performed directly on the raw event streams. All models are trained using the Adam optimizer with a cosine annealing learning rate scheduler. The initial learning rate is set to 1×10^{-3}, the minimum learning rate to 1×10^{-4}, and the weight decay to 1×10^{-4}. Each model is trained for 500 epochs with a batch size of 32. All models are trained using a single NVIDIA RTX 3090 GPU.

Methods	CIF10	N-C	N-M	ASL	GesT
EST	63.4	91.9	99.0	97.9	-
Matrix-LSTM	63.1	92.7	98.6	98.0	-
MVF-Net	68.7	92.7	98.1	97.1	-
EV-VGCNN	67.0	95.3	99.4	98.3	95.9
VMV-GCN	69.0	93.2	99.5	98.9	97.5
EVSTr	73.1	94.1	-	99.7	-
SlideGCN	68.0	93.1	99.1	-	-
EDGCN	71.6	**95.8**	-	98.5	98.5
VMST-Net	75.3	94.4	-	99.8	-
ASGCN	59.4	93.9	97.7	-	-
EVA	-	91.6	-	-	96.9
S3Net (this work)	**76.7**	95.6	**99.6**	**99.9**	**99.3**

Table 2: Comparison of methods on multiple event-based classification benchmarks.

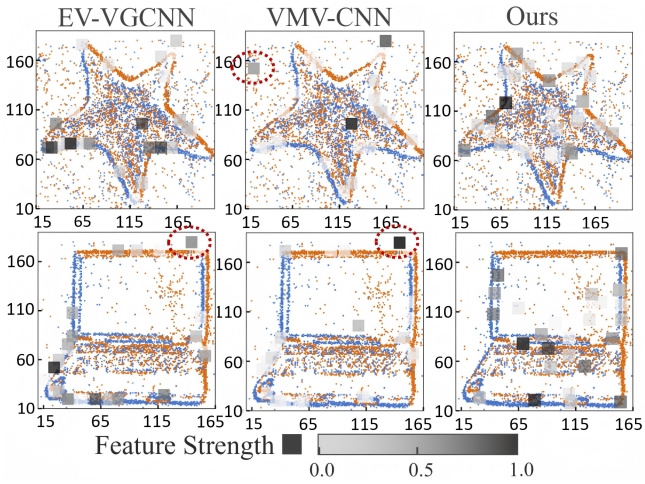

Figure 3: Feature Strength visualization of three voxel encoding strategies: EV-VGCNN, VMV-GCN, and our sparse encoding.

Experimental Results

Performance and Efficiency. To evaluate the computational efficiency and accuracy of our proposed method, we compare S3Net with a series of representative approaches on the N-Caltech101 dataset, as shown in Table 1. S3Net achieves the highest top-1 accuracy of 83.5%, surpassing the previous best result of 81.7% achieved by EST (Gehrig et al. 2019). Meanwhile, our model maintains a relatively low parameter count of 5.88 million, which is over three times smaller than MVF-Net (Deng, Chen, and Li 2021), and achieves the fastest inference time of 7.3 milliseconds, significantly outperforming EV-VGCN (Deng et al. 2022) and Matrix-LSTM (Cannici et al. 2020), which require 14.8 ms and 16.6 ms, respectively. These results highlight the superior balance between accuracy and efficiency offered by our architecture.

A key advantage of S3Net lies in its support for asynchronous inference, allowing it to process event streams in real time without waiting for the entire input sequence. This property aligns well with the inherent characteristics of DVS. While several existing methods, such as AsyNet (Messikommer et al. 2020), AEGNN (Schaefer, Gehrig, and Davide 2022), and SlideGCN (Li et al. 2021), also adopt asynchronous processing paradigms, S3Net achieves the highest classification accuracy among them, validating its effectiveness in both performance and efficiency. In contrast to point-based GNNs like EV-VGCN (Deng et al. 2022) and EVSTr (Xie et al. 2024), which rely on hand-crafted relation modeling techniques such as K-Nearest Neighbor (KNN) graph construction and Farthest Point Sampling (FPS), our approach employs a fully end-to-end design. Through voxel-based discretization and sparse convolutions, we eliminate the need for explicit relational reasoning while maintaining scalability and speed.

We also estimate the overall computational complexity of S3Net by accounting for its sparse operations. Sparse convolutions contribute approximately *2P* FLOPs, where P is the number of valid input-output index pairs. For each BatchNorm and activation layer, the complexity is estimated as *3MC* FLOPs. Compared to dense convolutional models such as EST and Matrix-LSTM, our sparse design avoids redundant operations and reduces memory consumption, especially in low-activity regions, thus providing a more efficient solution for event-based stream processing.

Generalization Across Benchmarks. To further examine the generalization capability of our method, we evaluate S3Net on five diverse event-based classification datasets. As summarized in Table 2, S3Net achieves top-1 accuracy of 76.7% on CIFAR10-DVS, 99.6% on N-MNIST, 99.9% on ASL-DVS, and 99.3% on DVSGesture. On N-CARS, it ranks second with an accuracy of 95.6%, closely approaching the best-performing method at 95.8%. These consistent results across datasets with varying spatial resolutions, temporal dynamics, and task granularities indicate the robustness and generalization of our framework.

Ablation Studies

To better understand the effectiveness of each component in S3Net, we conduct a series of ablation studies on the N-Caltech101 dataset. Specifically, we analyze the impact of the sparse encoding strategy, the spatial block, and the temporal block.

Sparse Encoding. We compare our proposed learnable binary encoding with two widely used methods: EV-VGCNN and VMV-GCN. While EV-VGCNN and VMV-GCN achieve 80.80% and 80.32% accuracy, respectively,

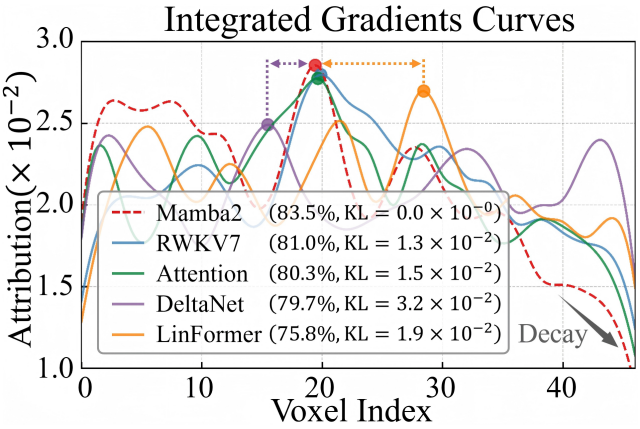

Figure 4: Comparison of Integrated Gradients across different temporal modeling methods.

our method attains 83.5%, demonstrating its superior ability to capture discriminative spatiotemporal features from asynchronous events. To further examine the effectiveness of the encoding schemes, we visualize the top-30 voxel features from each method in Figure 3. We observe that EV-VGCNN and VMV-GCN occasionally assign high feature responses to noisy event voxels irrelevant to object structure, as highlighted by the red dashed circles. In contrast, our method focuses more consistently on meaningful regions such as object contours and shape corners, reflecting enhanced polarity preservation and semantic alignment.

Voxel Spatial Block. We also conduct an ablation study on the VSB to evaluate the impact of its dual-branch design, as shown in Table 3. Using only the base branch yields an accuracy of 82.3%, while the motion branch alone achieves 81.9%. In contrast, the complete dual-branch configuration achieves higher accuracy than both single-branch variants. In terms of efficiency, the full configuration contains fewer parameters (5.88M) and lower computational complexity ($10.75C^2$) compared to the base branch (8.49M, $28C^2$) and the motion branch (6.22M, $13C^2$), where C denotes the number of channels. This efficiency gain results from using $0.5C$ channels in each branch, reducing the per-layer convolutional cost from C^2 to $0.25C^2$. Although two branches are used in parallel, the total parameter count and computational cost remain lower. These results indicate that the dual-branch design improves accuracy while maintaining lower complexity.

Voxel Temporal Block. To evaluate the effectiveness of different temporal modeling strategies within the VTB, we replace the default Mamba2 module with several representative architectures, including Attention (Vaswani et al. 2017), RWKV7 (Peng et al. 2025), DeltaNet (Yang et al. 2024), and LinFormer (Wang et al. 2020). Among these, Attention is evaluated using the same variable-length voxel sequence input as Mamba2. For models that do not support variable-length inputs (i.e., DeltaNet, RWKV7, and LinFormer), we employ a padding-based alignment strategy to convert all sequences to a fixed length.

Type	Cp. (C^2)	Params (M)	Acc. (%)
Base Block	28	8.49	82.3%
Motion Block	13	6.22	81.9%
Both	**10.75**	**5.88**	**83.5%**

Table 3: Ablation study comparing different block configurations in terms of computational complexity, model size, and accuracy.

We adopt the Integrated Gradients (IG) method (Sundararajan, Taly, and Yan 2017) to visualize the contribution of each temporal modeling module to the final classification decision. As shown in Figure 4, we compute the attribution score of each voxel in the final output layer of the VTB and plot the resulting curves according to voxel indices. Across all models, attribution scores tend to decrease along the temporal axis, indicating that voxels in the later part of the sequence play a relatively smaller role in the decision process. We mark the voxel with the highest attribution score for each model using a circular indicator. A clear correlation is observed between the attribution peak position and classification accuracy, where models with peaks closer to that of Mamba2 tend to perform better. For instance, RWKV7 achieves 81.0% accuracy with a peak near Mamba2's, while LinFormer, with the largest deviation, yields the lowest accuracy at 75.8%. This trend is further supported by the Kullback–Leibler (KL) divergence between attribution distributions, where smaller divergence values consistently correspond to better performance, reinforcing the effectiveness of Mamba2 in capturing essential temporal dependencies and discriminative voxel features.

Conclusion

In this work, we propose a spatiotemporally decoupled sparse encoding framework tailored for efficient event stream processing. This method combines high accuracy with low computational complexity, benefiting from a modular design that supports scalable modeling of asynchronous, sparse inputs. Extensive evaluations on multiple event-based classification datasets demonstrate its effectiveness. Future directions include extending the framework to more complex tasks such as event-based detection and segmentation, as well as incorporating adaptive voxelization and task-driven temporal modeling to further improve versatility and robustness.

Acknowledgements

This work was supported by the Natural Science Foundation of Sichuan (Grant No. 2024NSFSC1470 and 24NSFSC3404) and National Major Scientific Instruments and Equipments Development Project of National Natural Science Foundation of China (Grant No. 62427820).

References

Abadal, S.; Jain, A.; Guirado, R.; López-Alonso, J.; and Alarcón, E. 2021. Computing graph neural networks: A survey from algorithms to accelerators. *ACM Computing Surveys (CSUR)*, 54(9): 1–38.

AliAkbarpour, H.; Moori, A.; Khorramdel, J.; Blasch, E.; and Tahri, O. 2024. Emerging Trends and Applications of Neuromorphic Dynamic Vision Sensors: A Survey. *IEEE Sensors Reviews*.

Amir, A.; Taba, B.; Berg, D.; Melano, T.; McKinstry, J.; Di Nolfo, C.; Nayak, T.; Andreopoulos, A.; Garreau, G.; Mendoza, M.; et al. 2017. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7243–7252.

Benosman, R.; Clercq, C.; Lagorce, X.; Ieng, S.-H.; and Bartolozzi, C. 2013. Event-based visual flow. *IEEE transactions on neural networks and learning systems*, 25(2): 407–417.

Bertasius, G.; Wang, H.; and Torresani, L. 2021. Is space-time attention all you need for video understanding? In *ICML*, volume 2, 4.

Bi, Y.; Chadha, A.; Abbas, A.; Bourtsoulatze, E.; and Andreopoulos, Y. 2019. Graph-based object classification for neuromorphic vision sensing. In *Proceedings of the IEEE/CVF international conference on computer vision*, 491–501.

Bi, Y.; Chadha, A.; Abbas, A.; Bourtsoulatze, E.; and Andreopoulos, Y. 2020. Graph-based spatio-temporal feature learning for neuromorphic vision sensing. *IEEE Transactions on Image Processing*, 29: 9084–9098.

Cannici, M.; Ciccone, M.; Romanoni, A.; and Matteucci, M. 2020. A differentiable recurrent surface for asynchronous event-based data. In *European Conference on Computer Vision*, 136–152. Springer.

Dao, T.; and Gu, A. 2024. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*.

Deng, Y.; Chen, H.; and Li, Y. 2021. MVF-Net: A multi-view fusion network for event-based object classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(12): 8275–8284.

Deng, Y.; Chen, H.; Liu, H.; and Li, Y. 2022. A voxel graph CNN for object classification with event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1172–1181.

Feng, Y.; Lv, H.; Liu, H.; Zhang, Y.; Xiao, Y.; and Han, C. 2020. Event density based denoising method for dynamic vision sensor. *Applied Sciences*, 10(6): 2024.

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

Ge, C.; Fu, X.; He, P.; Wang, K.; Cao, C.; and Zha, Z.-J. 2025. EventMamba: Enhancing Spatio-Temporal Locality with State Space Models for Event-Based Video Reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 3104–3112.

Gehrig, D.; Loquercio, A.; Derpanis, K. G.; and Scaramuzza, D. 2019. End-to-end learning of representations for asynchronous event-based data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5633–5643.

Graham, B.; and van der Maaten, L. 2017. Submanifold Sparse Convolutional Networks. *arXiv preprint arXiv:1706.01307*.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Hamilton, J. D. 1994. State-space models. *Handbook of econometrics*, 4: 3039–3080.

Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. 2022. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1): 87–110.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Khan, N.; Iqbal, K.; and Martini, M. G. 2020. Lossless compression of data from static and mobile dynamic vision sensors-performance and trade-offs. *IEEE Access*, 8: 103149–103163.

Lagorce, X.; Orchard, G.; Galluppi, F.; Shi, B. E.; and Benosman, R. B. 2016. Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(7): 1346–1359.

Li, H.; Liu, H.; Ji, X.; Li, G.; and Shi, L. 2017. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11: 244131.

Li, Y.; Zhou, H.; Yang, B.; Zhang, Y.; Cui, Z.; Bao, H.; and Zhang, G. 2021. Graph-based asynchronous event processing for rapid object recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 934–943.

Liu, D.; Wang, T.; and Sun, C. 2023. Voxel-based multi-scale transformer network for event stream processing. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(4): 2112–2124.

Liu, X.; Zhang, C.; and Zhang, L. 2024. Vision mamba: A comprehensive survey and taxonomy. *arXiv preprint arXiv:2405.04404*.

Martin-Turrero, C.; Bouvier, M.; Breitenstein, M.; Zanuttigh, P.; and Parret, V. 2024. Alert-transformer: Bridging asynchronous and synchronous machine learning for real-time event-based spatio-temporal data. *arXiv preprint arXiv:2402.01393*.

Messikommer, N.; Gehrig, D.; Loquercio, A.; and Scaramuzza, D. 2020. Event-based asynchronous sparse convolutional networks. In *European Conference on Computer Vision*, 415–431. Springer.

Orchard, G.; Jayawant, A.; Cohen, G. K.; and Thakor, N. 2015. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9: 437.

Peng, B.; Zhang, R.; Goldstein, D.; Alcaide, E.; Du, X.; Hou, H.; Lin, J.; Liu, J.; Lu, J.; Merrill, W.; et al. 2025. Rwkv-7" goose" with expressive dynamic state evolution. *arXiv preprint arXiv:2503.14456*.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qin, X.; Zhang, J.; Bao, W.; Lin, C.; and Chen, H. 2025. Event Vision Sensor: A Review. *arXiv preprint arXiv:2502.06116*.

Schaefer, S.; Gehrig, D.; and Davide, S. 2022. Aegnn: Asynchronous event-based graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12371–12381.

Sekikawa, Y.; Hara, K.; and Saito, H. 2019. Eventnet: Asynchronous recursive event processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3887–3896.

Sironi, A.; Brambilla, M.; Bourdis, N.; Lagorce, X.; and Benosman, R. 2018. HATS: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1731–1740.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, 3319–3328. PMLR.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, S.; Li, B. Z.; Khabsa, M.; Fang, H.; and Ma, H. 2020. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

Xie, B.; Deng, Y.; Shao, Z.; Liu, H.; and Li, Y. 2022. Vmv-gcn: Volumetric multi-view based graph cnn for event stream classification. *IEEE Robotics and Automation Letters*, 7(2): 1976–1983.

Xie, B.; Deng, Y.; Shao, Z.; Xu, Q.; and Li, Y. 2024. Event voxel set transformer for spatiotemporal representation learning on event streams. *IEEE Transactions on Circuits and Systems for Video Technology*.

Xu, H.; Liu, Z.; Fu, R.; Su, Z.; Wang, Z.; Cai, Z.; Pei, Z.; and Zhang, X. 2024. PackMamba: Efficient Processing of Variable-Length Sequences in Mamba Training. In *European Conference on Computer Vision*, 34–42. Springer.

Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10): 3337.

Yang, N.; Wang, Y.; Liu, Z.; Li, M.; An, Y.; and Zhao, X. 2025. SMamba: Sparse Mamba for Event-based Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 9229–9237.

Yang, S.; Wang, B.; Zhang, Y.; Shen, Y.; and Kim, Y. 2024. Parallelizing linear transformers with the delta rule over sequence length. *Advances in neural information processing systems*, 37: 115491–115522.