# EvAug: Integrating Hierarchical and Adaptive Spatio-Temporal Augmentations into Event-Based Data by Mimicking Real-World Object Patterns

**Yukun Tian**
School of Computer Science
Southeast University

**Yongjian Deng**
College of Computer Science
Beijing University of Technology

**Wei You**
Advanced computing and storage Lab
Huawei Technologies Co., Ltd,

**Hao Chen***
School of Computer Science
Southeast University

## Abstract

Event cameras have shown great potential in various applications due to their low latency and high dynamic range. However, challenges such as data scarcity and limited diversity hinder model generalization, and research on event-specific data augmentation remains limited. This work aims to address this gap by introducing a systematic augmentation scheme named *EvAug*, which is inspired by real-world object patterns. In particular, we first propose *Multi-scale Temporal Integration* (MSTI) to diversify the motion speed, then introduce *Spatial-Salient Event Mask* (SSEM) and *Temporal-Salient Event Mask* (TSEM) to enrich object variants by emulating occlusion and interruption. Our EvAug can facilitate models learning with richer motion patterns, object variants and local spatio-temporal relations, thus improving model robustness and generalization capabilities. Experiment results demonstrate that EvAug consistently yields significant improvements across different tasks, representations, backbones and in multi-modal scenarios (*e.g.,* 4.87% accuracy gain on gesture recognition, 12.03% gain on object classification and 1.8% mIOU gain on semantic segmentation).

*Corresponding Author

Preprint. Under review.

# 1 Introduction

Event camera, which is also known as Dynamic Vision Sensors(DVS), is a new kind of bio-inspired device that differ fundamentally from conventional RGB frame cameras. Instead of capturing absolute brightness values, event cameras focus solely on changes in brightness, thus it has several unique features, including low-latency, low energy consumption, and exceptionally high dynamic range. These advantages make event camera a powerful tool in research areas like recognition [1, 2], depth estimation [3, 4], optical flow estimation [5, 6], and segmentation [7, 8].
However, event data is inherently sparse and limited in quantity, making annotation challenging and resulting in a lack of high-quality labeled datasets. Compared to their RGB counterparts for the same tasks, event-based datasets are often significantly smaller. For example, CIFAR-10DVS [9] is much smaller than CIFAR-10 [10] for recognition, and

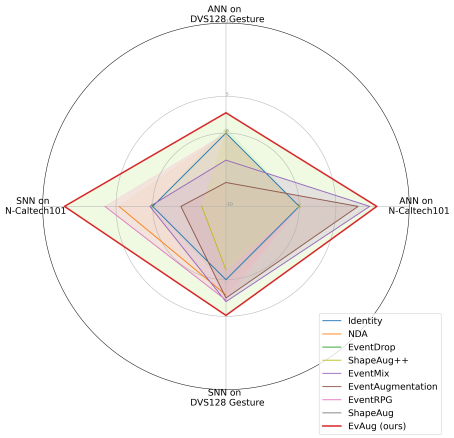


Figure 1: Comparison of our EventAug and other state-of-the-art augmentation methods.

DSEC-Semantics [11] is notably smaller than ADE20K [12] or SA-1B [13] for segmentation. This scarcity leads to severe overfitting and insufficient feature learning, as illustrated in Figure 2, thereby limiting the broader application of event camera.

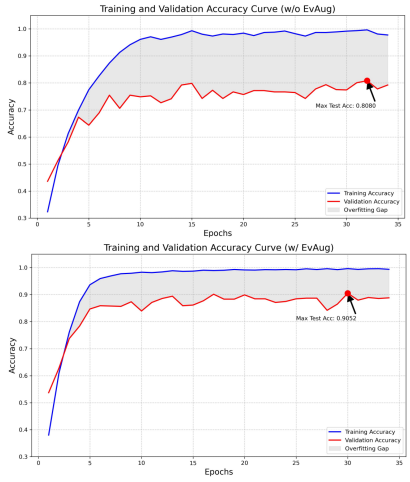


Figure 2: Training/test accuracy curves on N-Caltech101 dataset w/ and w/o EvAug.

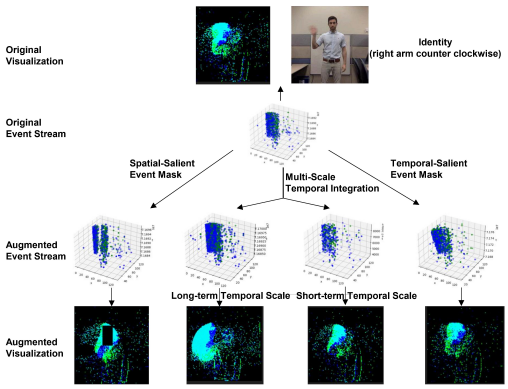


Figure 3: Examples of augmented events with EvAug, including original and augmented event stream, using event frame for visualization.

To improve performance, existing works mainly focus on backbone design and task-specific network building. Representative works include the Group Event Transformer [14], Spikepoint [15], video deraining [16], and motion deblurring [17]. However, these methods lack general applicability across diverse tasks, model architectures, and event representations. Hence, developing universally effective approaches to mitigate overfitting and boost performance across various scenarios remains a critical research priority in the event-based learning community.

Data augmentation is a practical method to solve the problems above: **It can improve model performance in a parameter-free manner without adding inference latency, which is friendly to the real-world application and deployment.** However, due to the sparse and non-uniform nature of event data, designing highly efficient augmentation policies remains a challenging task. As a result, research on event data augmentation has been relatively limited. Currently, existing event augmentation strategies can be broadly categorized into two main approaches.

*(i)* Directly transferring the conventional data augmentation methods for images to the event frames [18]. These augmentations apply the paradigms designed for RGB modality, ignoring

the sparsity and temporal characteristics of event data, resulting in limited augmentation performance. Moreover, they can only be performed on event frame, lacking generalization across different representations. *(ii)* Utilizing the temporal information of event data in a coarse manner. Representative work includes randomly dropping and mixing events [19, 20]. These methods rely heavily on randomness and prior assumptions (*e.g.,* uniform or Gaussian distribution) which ignores the non-uniform spatio-temporal distribution of event data. Thus, the result will be considerably poor when augmenting irrelevant spatial or temporal area. More importantly, they overlook modeling real-world object patterns, lacking physical interpretability and limiting the augmentation effectiveness. In addition, some methods are designed for specific networks or tasks, lacking generalization ability across different scenarios.

Given the limitations of existing methods, we argue that ideal event data augmentations should leverage rich spatio-temporal information, address the sparsity and non-uniform distribution and reasonably improve the model's generalization capabilities across various scenarios (Table 1).

To this end, we propose a systematic augmentation scheme named *EvAug* to solve the problems above. It contains three novel spatio-temporal augmentation methods: *Multi-scale Temporal Integration(MSTI)*, *Spatial-Salient Event Mask(SSEM)* and *Temporal-Salient Event Mask(TSEM)* (Figure 3). **Our methods aim at fully utilizing the rich spatio-temporal information inside event data and enhance the diversity of training samples by mimicking real-world object patterns**. This design ensures strong physical interpretability (e.g., motion speed) and encourages the generated samples to align closely with the true distribution of event data.

Table 1: A summary of event augmentation methods and evaluation of whether each method has experimentally proven effectiveness on various backbones (SNN&ANN), tasks, event representations, and multi-modal scenarios. Our EvAug shows comprehensive improvements across all settings.

| Augmentation Methods | Backbone Types | Tasks | Representations | Multi-Modal Scenarios |
|---|---|---|---|---|
| EventDrop [19] | × | × | ✓ | × |
| NDA [18] | × | × | × | × |
| EventMix [20] | ✓ | ✓ | × | × |
| ShapeAug [21] | × | ✓ | × | × |
| ShapeAug++ [22] | × | × | × | × |
| EventAugmentation [23] | ✓ | ✓ | ✓ | × |
| EventRPG [24] | × | × | × | × |
| **EvAug (Ours)** | ✓ | ✓ | ✓ | ✓ |

**For MSTI, we enrich the diversity of motion patterns by mimicking diverse motion speed of real-world scenarios.** Precisely, by leveraging the temporal integration mechanism, we adjust the scale of integration to make model simultaneously learn from multi-scale temporal context. By applying this augmentation, we efficiently generate samples under diverse motion speeds from single event stream by changing the number of event in representation procedure. Features such as edges and motions also change with the integration scale, thereby enhancing feature diversity and making the network more robust to diverse motion patterns. (Figure 4 left). Also, MSTI can boost model robustness to noise since different integration scales possess diverse noise levels.

**For SSEM and TSEM, we employ saliency-guided spatial and temporal mask to mimic occlusion and motion interruption and diversify local spatial and temporal correlations.** Masking encourage models to learn more spatio-temporal relationships from the unmasked regions, thereby enhancing feature learning. Specifically, to address the non-uniform spatial-temporal distribution of event data, we propose a fast, training-free method to obtain spatial and temporal saliency. Compared to previous methods with strong prior assumptions [19, 20], with the guidance of saliency information, we only perform augmentation in regions of interest, making our methods highly effective and adaptive to different datasets distribution. (Figure 4 right)

With the proposed design, our *EvAug* addresses the key limitations of existing methods (*e.g.,* limited scenario, prior assumption bias, lack of interpretability). It consistently achieves state-of-the-art performance across various tasks, datasets, network architectures, and event representations (Figure 1). In summary, our main contributions are as follows:

(1) We propose the Multi-scale Temporal Integration (MSTI) technique to enhance the diversity of motion patterns by mimicking objects in various motion speeds. MSTI allows model to learn

additional motion cues and spatial features from mutiple temporal context, providing model with enhanced generalization capabilities and induces model to focus on key discriminative regions.

(2) We introduce Spatial-Salient Event Mask (SSEM) and Temporal-Salient Event Mask (TSEM) to diversify local correlations by mimicking occlusion and motion interruption. These methods address the non-uniform spatial and temporal distribution of event data by utilizing spatial and temporal saliency, thus efficiently improve the diversity of local spatio-temporal correlations, and enhance robustness to occlusion and motion interruption in complex scenarios.

(3) Experimental results on various tasks, backbones, representations and multi-modal scenarios demonstrate that our proposed methods result in significant enhancements in performance and much better generalization ability over previous works. This also validates the idea that simulating real-world object patterns in event domain serves as an effective approach to event data augmentation, offering valuable insights for future studies. Our code will be publicly available for this community.

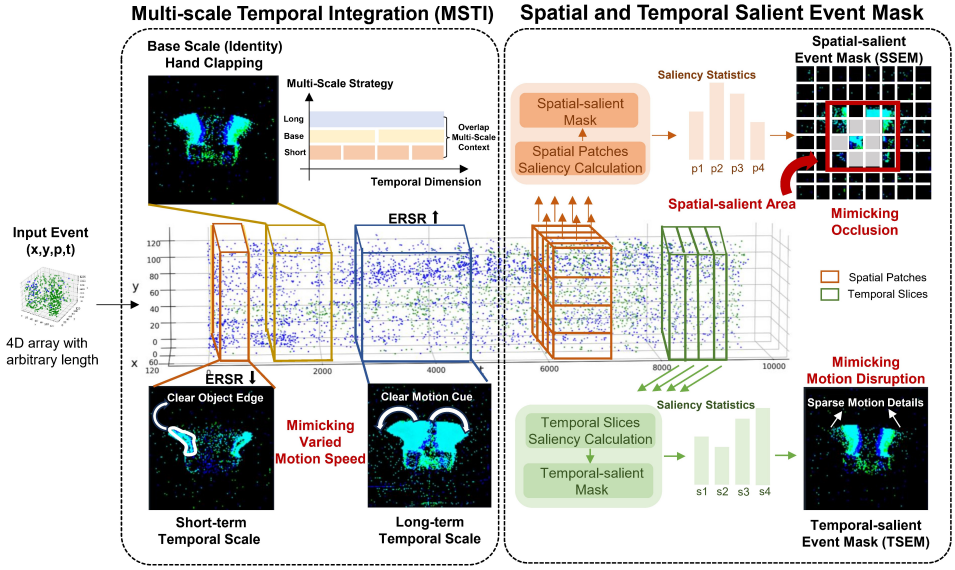


Figure 4: Illustration of our *EvAug* methods. The left is MSTI. By applying a multi-scale integration strategy, we enable the model to better learn motion information together with edge feature in multi-scale context. The right is SSEM and TSEM. Guided by the saliency information, we selectively mask event in salient spatial patches and salient temporal slices.

## 2 Related work

**Event-based Learning.** Recently, event-based learning has become a popular research area due to the development of Dynamic Vision Sensor and neuromorphic computing [25, 26]. Existing work in event community mainly focuses on backbone design and task-specific network building. Some representative works include Group Event Transformer [14], Spikepoint [15], video deraining [16] and motion deblurring [17]. However, the limited amount of event data and the large variation between event datasets restrict the performance of the network and ultimately lead to poor model generalization. In order to overcome these difficulties, many learning strategies have been proposed, such as unsupervised learning [16], self-supervised learning [27], pre-training and transfer learning [28, 29]. However, they commonly rely heavily on the paired RGB data which is hard to acquire, and many of them can not be generalized to other tasks.

Therefore, there is an urgent need for an efficient, task-agnostic method to overcome data deficiency. Our work aims to address this long-standing critical challenge within the community by introducing a systematic augmentation scheme to overcome limitations of existing methods.

**Event Data Augmentation.** Data augmentation always plays an important role in enhancing the models' generalization ability [30, 31, 32]. But for event data augmentation, only a small amount of work exists. Representative work includes: NDA [18], which applies geometry transformation of RGB modality like CutMix [33] and flip onto event, does not consider the unique sparse and spatio-

temporal nature of event data, thus achieve limited performance. EventDrop [19], which designs 3 kinds of random dropout strategies to improve the diversity of original datasets, and EventMix [20], which applies a three-dimensional version of Mixup [34] and CutMix [33] on the event data. Although they take both the spatial and temporal dimension into account, their augmentation performance is also unsatisfactory and unstable since they relies heavily on random and prior assumptions, ignoring the non-uniform and diverse distributions of different event data. More importantly, they overlook modeling real-world object patterns, lacking physical interpretability and limiting the augmentation effectiveness. In addition, some methods are designed for specific scenarios, lacking generalization ability. For instance, EventRPG [24] use Spiking Layer-Time-wise Relevance Propagation to perform mix and drop, which is designed for SNNs on classification, and not suitable for ANNs.

In contrast, *EvAug* is tailored for event data, addresses the key limitations of existing methods (*e.g.,* limited scenarios, prior assumption bias, lack of interpretability).

# 3 Method

## 3.1 Overview

The focus of *EvAug* is to efficiently perform spatio-temporal data augmentations to event data via mimicking real-world object patterns. We simulate physical phenomena (e.g., occlusion, speed variation) in the event domain to generate reasonable augmented samples that align with the natural distribution of event data and enhance model robustness under various real-world scenarios. Also, augmented samples contain richer and diverse temporal and spatial features than unaugmented data, thus enhance model learning. Specifically, this includes three methods: *Multi-scale Temporal Integration (MSTI)*, *Spatial-Salient Event Mask (SSEM)* and *Temporal-Salient Event Mask (TSEM)*.

## 3.2 Preliminary: Dense Event Representation

Event data is typically a 4-D set with arbitrary length. Due to sparsity of event data, mainstream deep learning approaches choose to transform event into dense representation as input for further feature extraction (e.g., voxel grid [35], event frame [36]). We will introduce the details in the following:
Let $E$ denotes the sequence of an event stream, $N$ is the total number of event:

$$E = \{E_i\}_{i=1}^{N} = \{(x_i, y_i, p_i, t_i)\}_{i=1}^{N} \tag{1}$$

$(x_i, y_i)$ is the coordinate where the event $E_i$ generates, $t_i$ is the timestamp indicates when the event is generated, and $p_i$ is the polarity with 1 and -1 indicating positive and negative events respectively. We pre-arrange the event stream in timestamp order. To generate dense representation, we commonly divide the event stream into $T$ slices. Let $R(j) \in \mathbb{R}^{M \times H \times W}$(M is determined by the representation methods) denotes the representation that generates from the $j$th event slice, $i_{start}$ and $i_{end}$ as the start and end timestamp for it. The slice window is typically determined by either a fixed number of events or a fixed temporal interval. Then, we perform temporal integration in the target time region:

$$R(j)_{x_i, y_i, p_i} = \sum_{k=i_{start}}^{i_{end}} f_{p\pm}(x_k, y_k, p_k, t_k)\mathbb{I}(E_k) \tag{2}$$

$$\mathbb{I}(E_k) = \begin{cases} 1, & x_k = x_i, y_k = y_i, p_k = p_i \\ 0, & otherwise \end{cases} \tag{3}$$

where $\mathbb{I}$ is the indicator function which ensure only events in target slice join the integration. $f_{\pm}$ is a measurement function, which assign weighted value to events based on the representation method. For example, $f_{\pm}(x_k, y_k, p_k, t_k) = \pm 1$ for event frame. After integration, each event stream transforms into $T$-channels dense event representation, and representation in each channel can be treated as an image with a resolution of [$W$, $H$].

## 3.3 Multi-scale Temporal Integration

Our *MSTI* aims at simulating objects under different motion speed. This method is inspired by our observations that motion speed determines the completeness of motion cues and the clarity of object boundaries within the same event representations. For fast-moving objects, more motion information is revealed including moving orbit, the speed of movement and so on. For slow-moving objects, more

information about the object itself is revealed (*e.g.,* contours, shapes). These can be easily discerned from the visualization (Figure 4 left).

Therefore, our key challenge is *How to generate objects under different motion speed in event domain?*. For event camera, each event is generated by equation:

$$logI(x_i, y_i, t_i + \Delta t) - logI(x_i, y_i, t_i) \geq p_i * \alpha \tag{4}$$

where $I(x, y, t)$ represents the absolute illumination at position $(x, y)$ and time $t$, $\alpha$ is the threshold of the event camera, and $p_i \in \{-1, +1\}$ denotes the polarity. Under identical lighting conditions, the same object moving along a fixed trajectory will induce highly similar illumination changes. Moreover, the trigger of events depends on the magnitude of brightness change rather than the exact timing $t_i$. Therefore, when we decompose fast and slow motions into $K$ minimal motion units that are sufficient to trigger events, i.e., $M = \{m_i\}_{i=1}^{K}$, then—under ideal conditions (i.e., no refractory period for event camera, sufficient temporal resolution and bandwidth)—both fast and slow motions should produce a similar number of events. **Thus, the key distinction between motions at different speeds lies in the *number of events triggered per unit time*, rather than the total number of events generated throughout the entire motion.** Former work defines this as *event rate* [37]. However, dense representations discretize event stream into slices, making event rate invalid. To address this, we introduce *Event Representation Slice Rate (ERSR)*, which quantifies the number of events contained in each representation slice.

$$ERSR_i = \frac{\sum_{k=1}^{N} \mathbb{I}'(E_k)}{N}, \qquad \mathbb{I}'(E_k) = \begin{cases} 1, & t_k \in [i_{start}, i_{end}] \\ 0, & otherwise \end{cases} \tag{5}$$

This observation implies that we can emulate different object motion speeds by adjusting *ERSR* involved in the integration process, i.e., by adjusting the temporal integration scale of the event stream. This allows us to generate augmented samples in a physically meaningful manner. Taking fixed temporal interval integration as example, we adopt a speed-aware policy, which include three different scale: Selecting both $\frac{1}{n}$ (short-term temporal scale) and the m-fold (long-term temporal scale) of the base *ERSR* (n and m are hyper parameters), together with the base *ERSR* to generate event representation. For a single event stream, we can now generate samples under three different motion speed, making model simultaneously learn feature from multi-scale temporal contexts.

Also, representations with different integration scales contain varying degrees of noise which is beneficial for feature extraction as it induce model to focus on discriminative features rather than background noise (supportive experiment in 4, Figure 5). This augmentation policy is highly general and can be applied to various representations, tasks and model architectures.

### 3.4 Spatial and Temporal Salient Event Mask

Our masking strategies are designed to simulate occlusions and motion interruptions—two particularly challenging scenarios for visual systems in real-world environments. By incorporating such examples during training, we aim to improve the model's robustness under these conditions. More importantly, inspired by [38], we argue that masking can also enrich local spatial and temporal correlations. By forcing the model to infer from unmasked regions, masking encourages better learning of spatio-temporal dependencies, thereby enhancing feature representation.

Considering the inherently non-uniform spatio-temporal distribution of event data, we further improve the effectiveness of our augmentation strategy by introducing saliency guidance. Specifically, we propose two saliency-aware methods: *Spatial-Salient Event Mask (SSEM)* and *Temporal-Salient Event Mask (TSEM)*. We first compute spatio-temporal saliency based on event density distributions along spatial and temporal dimensions. Masking is then selectively applied to salient spatial or temporal areas, mitigating the inherent sparsity and non-uniform distribution of event data.

**Spatial-Salient Event Mask** We first obtain the spatial saliency of event data by a fast and training-free method we propose in Algorithm 1. This approach leverages the unique sparse nature of event data, which fundamentally differs from the dense structure of RGB images. Therefore, we can efficiently acquire the spatial saliency map of the event sequence by calculating its density distribution along spatial dimension. With the sparse nature of event, the density distribution serves as an effective approximation of its semantics saliency map. Different from former saliency methods like [39, 40, 24], our saliency offering high computational efficiency **without requiring additional training or inference**. To elaborate, we first divide the event sequence into patches in spatial

dimensions $(x, y)$ like in [41], then we calculate the density distribution of event in each patch. The density distribution is then approximated as the spatial saliency map, as described in Algorithm 1.

**Algorithm 1** Spatial-saliency calculation

**Require:** $E$ for event stream, $P_i$ for patch $i$, one event slice have $k$ patches in total. $S$ denotes the patch saliency for each patch.
**Ensure:** $idx$ (the salient patches indexes)
  **Init**: S = [], idx = [0,1,$\cdots$,k-1]
  **for** $P_i \in Patch$ **do**
    **for** $E_j \in E$ **do**
      **if** $E_j \in P_i$ **then**
        $index \leftarrow j$
      **end if**
    **end for**
    $S[i] \leftarrow (len(index))$
  **end for**
  SORT($idx[i]$ based on $S[idx[i]]$ in descending order)

Given the spatial saliency information, we choose the most salient area of each sequence to apply event spatial mask, which mask out all the events in the target patches. First, we define $r$ as the mask rate. And we set a saliency threshold $\epsilon$, which is decided by the density of the target event stream and mask rate $r$, ensuring a percentage of $r$ patches are masked. Let $\mathbf{M}$ denotes the spatial-salient mask of a single event stream, $S_o$ is the original event sequence, $S_M$ is the sequence after augmentation, $p$ denotes all the patches in $S$,

$p_s$ is the salient patches we get from the algorithm 1, function $Dense(p_i)$ returns the event density of area $p_i$ by performing the algorithm above. The detailed masking method is as follows:

We first calculate the threshold of event density saliency $\epsilon$:

$$Idx = idx[kr - 1], \qquad \epsilon = Dense(x_{Idx}) \tag{6}$$

Then, we obtain the spatial-salient mask by determining whether each patch is salient:

$$\begin{cases} p_i \in p_s, & Dense(p_i) > \epsilon \\ p_i \notin p_s, & otherwise \end{cases}, \qquad \mathbf{M_{i,j}} = \begin{cases} 0, & (i,j) \in p_s \\ 1, & otherwise \end{cases} \tag{7}$$

Finally, we acquire the masked sequence $S_M$ by applying Hadamard product with original sequence $S_o$ and mask $\mathbf{M}$.

$$S_M = S_o \odot \mathbf{M} \tag{8}$$

Now we get the augmented sequence $S_M$.

**Algorithm 3** Temporal-Salient Event Mask

**Require:** $E$ for original event stream, $slice_i = [t_{start}, t_{end}]$ for the timestamp of the $i$th event frame
**Ensure:** idx (the salient slices indexes)
**Init**: idx = [0, 1, $\cdots$, T-1], count = 0
**function** TEMPORAL-SALIENCY CALCULATION
  **for** $slice_i \in slice$ **do**
    $count = 0$
    **for** $E_j \in E$ **do**
      **if** $E_j \in slice_i$ **then**
        $count \leftarrow count + 1$
      **end if**
    **end for**
    $s[i] \leftarrow count$
  **end for**
  SORT($idx[i]$ based on $s[idx[i]]$ in descending order)
**end function**

**Require:** $E$ for original event stream, $slice$ for the target salient event frame slice, $p$ for a base mask rate, $d = [d_1, d_2, \cdots, d_T]$ for the event density of the salient slice
**Ensure:** $E_m$ (event stream after masking)
**Init**: idx = [0, 1, $\cdots$, T], $index_M$ = []
**function** TEMPORAL-SALIENT EVENT MASK
  $m = \min(d)$
  **for** $i = 0$ to $T - 1$ **do**
    $p_s = d_i/m * p$
    $index = \{j \mid E_j \in slice_i\}$
    **for** $k$ in $index$ **do**
      **if** RANDOM(0,1) < $p_s$ **then**
        $index_m \leftarrow k$
      **end if**
    **end for**
    $index_M = index_m \cup index_M$
  **end for**
  $E_m = E \setminus E_{index_M}$
**end function**

**Temporal-Salient Event Mask** Similar to SSEM, we propose *TSEM* to diversify local temporal correlations by mimicking motion interruptions. Given that event data inherently encode temporal information, we extend the concept of spatial saliency to the temporal dimension and define *temporal saliency*, which approximately measures semantic relevance along the temporal axis. We compute temporal saliency via a similar density substitution mechanism as in SSEM that is both efficient and training-free. Specifically, we divide the event stream into $T$ temporal slices, where $T$ corresponds to the number of bins determined by the event representation policy described in Section 3.2. The detailed computation procedure for temporal saliency is presented on the left side of Algorithm 2.

For temporal masking, we first get the temporal saliency by algorithm above, then we perform a saliency-guided temporal mask. Inside the salient event slice, we first get the minimum density of the target salient event slice $m$, and we define a base mask rate $p$ which is a hyper parameter. The

mask rate $p_s$ of each target salient slice will be decided by their event density and base mask rate. For event $e$ in the slice, the mask probability of it is equal to $p_s$. We describe this method in Algorithm 2 right in detail.

# 4 Experiment

## 4.1 Gesture Recognition and Object Classification

**Implementation** Our experiments are conducted using Pytorch [42], with Adam [43] optimizer and a learning rate of 0.001. For our own implementation, We conducted multiple rounds of experiments and reported the median results. Detailed implementation information is provided in the supplementary materials. To assess the generalization of our methods, we evaluate the augmentation technique on two distinct deep neural network architectures: **(i) Spiking Neural Network (SNN)**, which are considered the most suitable network architecture for processing event data. We choose the convolution spiking neural network (CSNN) defined and implemented by [44] for evaluation, which contains 5 convolution layers, 2 full connection layers and one voting layer. The scale of the parameters is approximately 1.7M. **(ii) Artificial Neural Network(ANN)**. We follow former studies [45, 19] to use Resnet-34 [46]. The scale of the parameters is approximately 21.8M.

Table 2: Comparison of object classification and gesture recognition accuracy (%) of various augmentation methods on different datasets and backbones. * means running in our own environment. Numbers in parentheses indicate improvement over identity.

| Method | Model | DVS128 Gesture | N-Caltech101 |
|---|---|---|---|
| EventDrop* [19] | Resnet-34 (ANN) | 96.18 | - |
| EventMix [20] | Resnet-34 (ANN) | 91.80 | 89.20 |
| EventAugmentation [23] | Resnet-34 (ANN) | 88.75 | 87.61 |
| Identity* | Resnet-34 (ANN) | 95.49 | 79.58 |
| EvAug (Ours)* | Resnet-34 (ANN) | **97.92** (**+2.43**) | **90.16(+10.58)** |
| EventDrop* [19] | CSNN (SNN) | 94.44 | - |
| NDA* [18] | CSNN (SNN) | 95.83 | - |
| NDA [18] | VGG11 (SNN) | - | 83.70 |
| EventMix [20] | Resnet-18 (SNN) | 96.75 | 79.47 |
| ShapeAug [21] | Resnet-34 (SNN) | 91.70 | 68.70 |
| ShapeAug++ [22] | Resnet-34 (SNN) | 92.40 | 72.40 |
| EventAugmentation [23] | CSNN (SNN) | 96.25 | 75.25 |
| EventRPG [24] | Resnet-18 (SNN) | 96.53 | - |
| EventRPG [24] | VGG11 (SNN) | - | 85.62 |
| Identity* | CSNN (SNN) | 93.75 | 79.10 |
| EvAug (Ours)* | CSNN (SNN) | **98.62** (**+4.87**) | **91.13(+12.03)** |

**Datasets** We follow previous works [20, 18, 19] to use two challenging public datasets N-Caltech101 [47] and DVS128 Gesture [48] for evaluation. For N-Caltech101, we follow [20, 21, 18] to divide the training and test sets by 9 : 1. For DVS128 Gesture, we follow [20, 44, 49] to divide the training and test sets by 8:2.

**Results** Table 2 compares our *EvAug* with other sota augmentation methods across different backbone architectures. *EvAug* consistently achieves significant performance improvements, showcasing its efficacy in enriching data diversity. Former methods like [18, 19] either neglects the sparse and spatio-temporal aspects of event or fails to address non-uniform distribution, leading to limited augmentation effects. In contrast, *EvAug* efficiently utilizes the sparse and spatio-temporal nature of event data, significantly boosting dataset diversity by capturing a broader spectrum of motion patterns and spatial-temporal correlations. More importantly, our proposed *EvAug* also serves as an effective regularization technique, significantly alleviating overfitting issues, as also evidenced in Figure 2.

## 4.2 Semantic Segmentation

**Implementation** We employ Segformer [50] and EISNet [8] as backbones. To evaluate the generalization of our *EvAug*, we consider different event representations, including Histogram [51], Voxel Grid [35], and Activity-Aware Event Tensor [8]. Our proposed augmentation framework, *EvAug*, is applied during training of event modal, details are in supplementary materials.

**Datasets** We evaluate our performance on DSEC-Semantics dataset [52, 11], a widely used benchmark for event and multi-modal semantic segmentation. We use the official train and test split.

**Results** Table 3 summarizes the semantic segmentation results on the DSEC-Semantics dataset, showcasing the performance improvements brought by our *EvAug*. For event-only inputs, our *EvAug* consistently outperforms the baseline (*Identity*) across all tested representations. These consistent gains across representations highlight the generalizability and effectiveness of *EvAug* in enhancing the feature learning process, overcoming the representation limitation in existing methods [18]. In the multi-modal scenario, *EvAug* delivers a substantial improvement in segmentation performance. This notable gain underscores the ability of *EvAug* to reveal unique information in event modal, indicating its potential to become a robust data augmentation technique for multi-modal event-based tasks.

### 4.3 Analysis and Discussion

**Visualization** We visualize the heatmap of models training with and without our augmentation. As shown in Figure 5 and analyzed in section 3.3, our *EvAug* can induce model to focus on key discriminative regions rather than background noise, enhancing feature learning and generalization.

**Efficiency** Table 4 presents the latency for computing saliency across synthetic and real datasets. TS and SS represent temporal and spatial saliency. The results suggests that our saliency computations are efficient and scalable.

**Simulating Scenarios Experiment** As shown in Table 5, we adjust the number of events per bin to simulate speed variations and apply a central mask (0.5H, 0.5W) to simulate occlusion. Note that these settings are held-out and do not appear in our augmented datasets. This suggest our *EvAug* can improve model robustness in challenging scenarios.

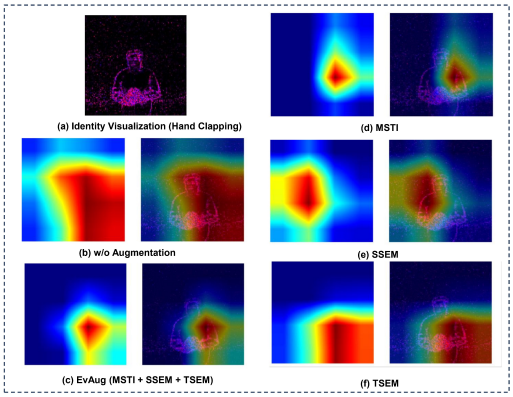


Figure 5: GradCam Visualization of our EvAug.

Table 4: Average latency **(ms)** of saliency calculation under different settings.

| Setting | Synthetic Data Length | | | Real Data |
|---|---|---|---|---|
| | 10,000 | 100,000 | 1,000,000 | DVS-Gesture |
| TS | 0.275 | 1.17 | 10.4 | 5.33 |
| SS | 0.972 | 7.49 | 135 | 38.5 |

Table 3: Semantic Segmentation performance of our EvAug on DSEC-Semantics datasets. E denotes only event, while E&I denotes multi-modal input of event and image.

| Model | Input | Rep | Aug | mIOU |
|---|---|---|---|---|
| Segformer | E | Histogram | Identity | 46.93 |
| Segformer | E | Histogram | Ours | **47.57** |
| Segformer | E | Voxel Grid | Identity | 46.21 |
| Segformer | E | Voxel Grid | Ours | **47.15** |
| Segformer | E | AEIM | Identity | 46.32 |
| Segformer | E | AEIM | Ours | **47.18** |
| EISNet | E&I | AEIM | Identity | 63.94 |
| EISNet | E&I | AEIM | Ours | **65.74** |

Table 5: Performance comparison (%) between the identity and augmented ResNet-34 on the DVS-Gesture dataset under different settings.

| Setting | Identity | EvAug |
|---|---|---|
| Original | 95.49 (-0.00) | 97.92 (-0.00) |
| 0.25× Speed | 80.56 (-14.93) | 91.67 (-6.25) |
| 4.0× Speed | 71.88 (-23.61) | 86.46 (-11.46) |
| Occlusion | 92.36 (-3.13) | 95.83 (-2.09) |

**Ablation Study** We conducted ablation experiments on the DVS128 Gesture dataset using two backbone networks to evaluate the effectiveness of our *EvAug* methods. Tables 6 demonstrates the impact of our three individual augmentation methods, showcasing its ability to enrich temporal diversity and better capture motion-related patterns, which are crucial for event-based data. In Table 7, we observe that ResNet-34, which emphasizes spatial feature extraction, benefits more from spatially-oriented augmentations such as *Spatial-Salient Event Mask (SSEM)*. Furthermore, when all three augmentation methods are combined, the accuracy reaches 98.26% (+2.77%). This

substantial improvement highlights the collaborative effect of jointly enhancing both spatial and temporal diversity in the dataset.

Table 6: Accuracy (%) of our proposed methods on DVS128 Gesture with various models.

| Model | Method | Accuracy (Gain) |
|---|---|---|
| Resnet-34 | Identity | 95.49(+0.00) |
| | MSTI | 96.53(+1.04) |
| | SSEM | 97.92(+2.43) |
| | TSEM | 96.53(+1.04) |
| CSNN | Identity | 93.75(+0.00) |
| | MSTI | 97.57(+3.82) |
| | SSEM | 96.18(+2.43) |
| | TSEM | 98.62(+4.87) |

Table 7: Performance of EvAug for Resnet-34 on DVS128 Gesture with different augmentation setting(%).

| Model | MSTI | SSEM | TSEM | Accuracy |
|---|---|---|---|---|
| Resnet-34 | × | × | × | 95.49(+0.00) |
| | ✓ | × | × | 96.53(+1.04) |
| | × | ✓ | × | 97.92(+2.43) |
| | × | × | ✓ | 96.53(+1.04) |
| | × | ✓ | ✓ | 97.57(+2.08) |
| | ✓ | ✓ | ✓ | **98.26(+2.77)** |

In supplementary materials, we also show that adding saliency to the masking strategy enhances augmentation, the efficacy of multi-scale strategy towards single scale, impacts of parts of hyperparameters and augmentation performance on low-level vision optical flow estimation task.

## 5 Conclusion

In this work, we introduce a spatio-temporal data augmentation method that diversifies motion speeds and local correlations using three strategies. *EvAug* alleviate overfitting, improves model robustness in challenging scenes and shows strong generalization across various settings. Our approach achieves significant improvements, as validated by experiments with multiple representations, backbones and tasks. In the future, we will expand this augmentation method to other event-based learning tasks and multi-modal data augmentation.

## References


[1] Yongjian Deng, Hao Chen, Hai Liu, and Youfu Li. A voxel graph cnn for object classification with event cameras. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1162–1171, 2022.

[2] Chang Liu, Xiaojuan Qi, Edmund Y. Lam, and Ngai Wong. Fast classification and action recognition with event-based imaging. *IEEE Access*, 10:55638–55649, 2022.

[3] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128$\times$ 128 120 db 15 $\mu$s latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008.

[4] Xu Liu, Jianing Li, Jinqiao Shi, Xiaopeng Fan, Yonghong Tian, and Debin Zhao. Event-based monocular depth estimation with recurrent transformers. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(8):7417–7429, 2024.

[5] Wachirawit Ponghiran, Chamika Mihiranga Liyanagedera, and Kaushik Roy. Event-based temporally dense optical flow estimation with sequential learning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9793–9802, 2023.

[6] Shintaro Shiba, Yannick Klose, Yoshimitsu Aoki, and Guillermo Gallego. Secrets of event-based optical flow, depth and ego-motion estimation by contrast maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):7742–7759, 2024.

[7] Timo Stoffregen, Guillermo Gallego, Tom Drummond, Lindsay Kleeman, and Davide Scaramuzza. Event-based motion segmentation by motion compensation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7243–7252, 2019.

[8] Bochen Xie, Yongjian Deng, Zhanpeng Shao, and Youfu Li. Eisnet: A multi-modal fusion network for semantic segmentation with events and images. *IEEE Transactions on Multimedia*, 26:8639–8650, 2024.

[9] Wensheng Cheng, Hao Luo, Wen Yang, Lei Yu, and Wei Li. Structure-aware network for lane marker extraction with dynamic vision sensor, 2020.

[10] Alex Krizhevsky. Learning multiple layers of features from tiny images. `https://www.cs.toronto.edu/~kriz/cifar.html`, 2009. (Updated 2019).

[11] Zhaoning Sun*, Nico Messikommer*, Daniel Gehrig, and Davide Scaramuzza. Ess: Learning event-based semantic segmentation from still images. *European Conference on Computer Vision. (ECCV)*, 2022.

[12] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.

[13] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[14] Yansong Peng, Yueyi Zhang, Zhiwei Xiong, Xiaoyan Sun, and Feng Wu. Get: Group event transformer for event-based vision. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6015–6025, 2023.

[15] Hongwei Ren, Yue Zhou, Yulong Huang, Haotian Fu, Xiaopeng Lin, Jie Song, and Bojun Cheng. Spikepoint: An efficient point-based spiking neural network for event cameras action recognition. *arXiv preprint arXiv:2310.07189*, 2023.

[16] Jin Wang, Wenming Weng, Yueyi Zhang, and Zhiwei Xiong. Unsupervised video deraining with an event camera. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10797–10806, 2023.

[17] Xiang Zhang, Lei Yu, Wen Yang, Jianzhuang Liu, and Gui-Song Xia. Generalizing event-based motion deblurring in real-world scenarios. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10700–10710, 2023.

[18] Yuhang Li, Youngeun Kim, Hyoungseob Park, Tamar Geller, and Priyadarshini Panda. Neuromorphic data augmentation for training spiking neural networks. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 631–649, Cham, 2022. Springer Nature Switzerland.

[19] Fuqiang Gu, Weicong Sng, Xuke Hu, and Fangwen Yu. Eventdrop: Data augmentation for event-based learning. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 700–707. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track.

[20] Guobin Shen, Dongcheng Zhao, and Yi Zeng. Eventmix: An efficient data augmentation strategy for event-based learning. *Information Sciences*, 644:119170, 2023.

[21] Katharina Bendig, René Schuster, and Didier Stricker. Shapeaug: Occlusion augmentation for event camera data. *ArXiv*, abs/2401.02274, 2024.

[22] Katharina Bendig, René Schuster, and Didier Stricker. Shapeaug++: More realistic shape augmentation for event data, 2024.

[23] Fuqiang Gu, Jiarui Dou, Mingyan Li, Xianlei Long, Songtao Guo, Chao Chen, Kai Liu, Xianlong Jiao, and Ruiyuan Li. Eventaugment: Learning augmentation policies from asynchronous event-based data. *IEEE Transactions on Cognitive and Developmental Systems*, 16(4):1521–1532, 2024.

[24] Mingyuan Sun, Donghao Zhang, Zongyuan Ge, Jiaxu Wang, Jia Li, Zheng Fang, and Renjing Xu. Eventrpg: Event data augmentation with relevance propagation guidance. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[25] Dongcheng Zhao, Yi Zeng, Tielin Zhang, Mengting Shi, and Feifei Zhao. Glsnn: A multi-layer spiking neural network based on global feedback alignment and local stdp plasticity. *Frontiers in Computational Neuroscience*, 14, 2020.

[26] Dongcheng Zhao, Yang Li, Yi Zeng, Jihang Wang, and Qian Zhang. Spiking capsnet: A spiking neural network with a biologically plausible routing rule between capsules. *Information Sciences*, 610:1–13, 2022.

[27] Simone Klenk, David Bonello, Lukas Koestler, and Daniel Cremers. Masked event modeling: Self-supervised pretraining for event cameras. *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2367–2377, 2022.

[28] Lin Wang, Yujeong Chae, and Kuk-Jin Yoon. Dual transfer learning for event-based end-task prediction via pluggable event to image translation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2115–2125, 2021.

[29] Yan Yang, Liyuan Pan, and Liu Liu. Event camera data pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10665–10675, 2023.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, may 2017.

[31] Teerath Kumar, Alessandra Mileo, Rob Brennan, and Malika Bendechache. Image data augmentation approaches: A comprehensive survey and future directions. 2023.

[32] Dipen Saini and Rahul Malik. Image data augmentation techniques for deep learning -a mirror review. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 1–5, 2021.

[33] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6022–6031, 2019.

[34] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018.

[35] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Live demonstration: Unsupervised event-based learning of optical flow, depth and egomotion. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1694–1694, 2019.

[36] Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. In *British Machine Vision Conference*, 2017.

[37] Manasi Muglikar, Siddharth Somasundaram, Akshat Dave, Edoardo Charbon, Ramesh Raskar, and Davide Scaramuzza. Event cameras meet spads for high-speed, low-bandwidth imaging, 2024.

[38] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, June 2022.

[39] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016.

[40] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[41] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[42] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[43] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[44] Wei Fang, Yanqi Chen, Jianhao Ding, Zhaofei Yu, Timothée Masquelier, Ding Chen, Liwei Huang, Huihui Zhou, Guoqi Li, and Yonghong Tian. Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence, 2023.

[45] Daniel Gehrig, Antonio Loquercio, Konstantinos Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5632–5642, 2019.

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[47] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015.

[48] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, Jeff Kusnitz, Michael Debole, Steve Esser, Tobi Delbruck, Myron Flickner, and Dharmendra Modha. A low power, fully event-based gesture recognition system. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7388–7397, 2017.

[49] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2641–2651, 2021.

[50] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: simple and efficient design for semantic segmentation with transformers. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2024. Curran Associates Inc.

[51] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1731–1740, 2018.

[52] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 2021.