GET: Group Event Transformer for Event-Based Vision

Yansong Peng Yueyi Zhang* Zhiwei Xiong Xiaoyan Sun Feng Wu
University of Science and Technology of China
pengyansong@mail.ustc.edu.cn, {zhyuey, zwxiong, sunxiaoyan, fengwu}@ustc.edu.cn

Abstract

Event cameras are a type of novel neuromorphic sensor that has been gaining increasing attention. Existing event-based backbones mainly rely on image-based designs to extract spatial information within the image transformed from events, overlooking important event properties like time and polarity. To address this issue, we propose a novel ***Group-based*** *vision Transformer backbone for Event-based vision, called Group Event Transformer (GET), which decouples temporal-polarity information from spatial information throughout the feature extraction process. Specifically, we first propose a new event representation for GET, named Group Token, which groups asynchronous events based on their timestamps and polarities. Then, GET applies the Event Dual Self-Attention block, and Group Token Aggregation module to facilitate effective feature communication and integration in both the spatial and temporal-polarity domains. After that, GET can be integrated with different downstream tasks by connecting it with various heads. We evaluate our method on four event-based classification datasets (Cifar10-DVS, N-MNIST, N-CARS, and DVS128Gesture) and two event-based object detection datasets (1Mpx and Gen1), and the results demonstrate that GET outperforms other state-of-the-art methods. The code is available at* https://github.com/Peterande/GET-Group-Event-Transformer.

1. Introduction

Event cameras are a type of bio-inspired vision sensors that capture per-pixel illumination changes asynchronously. Compared with traditional frame cameras, event cameras offer many merits like high temporal resolution (>10K fps), high dynamic range (>120 dB), and low power consumption (<10 mW) [20]. Many applications, such as object classification [39, 27] and high-speed object detection [2, 73, 47], can take advantage of this kind of camera, especially when power consumption is limited or in the presence of challenging motion and lighting conditions.

*Corresponding author

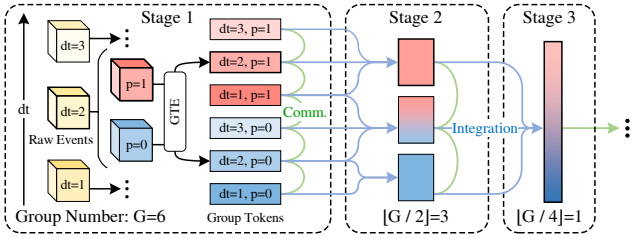


Figure 1. Group Tokens are generated in Stage 1 according to timestamps and polarities of events. In GET, these tokens are effectively communicated and integrated, making full use of important event information while maintaining information decoupling.

The event data are stored as asynchronous arrays, containing the location, polarity, and time of each illumination change. In contrast, traditional frames represent visual information as a matrix of pixel values captured at a fixed rate. As a result, traditional image-based neural networks can not be directly applied to event data. To address this issue, several works have been proposed to convert events into image-like representations. These representations, such as voxel grid [76], event histogram [46], and time surface [30, 55, 26], are then fed to deep neural networks [23, 18, 21, 38]. More recent works leverage the successful Transformer architecture to extract features directly from the above event representations [72, 62, 49, 24], or from the feature maps obtained by CNN-based backbones [56, 63, 75]. Although these networks have fair performance, they are still limited by their reliance on image-based feature extraction designs [16, 33, 25, 43]. These designs mainly extract spatial information and can not fully utilize the temporal and polarity information contained in events. There are also some works trying to bridge this gap. For example, Spiking neural networks (SNN) are utilized to capture temporal information by representing event data as spikes that occur over time and propagate through neurons [64, 6, 40]. Graph neural networks (GNN) are also adopted to model the dynamic interactions between nodes in an event graph over time [34, 51]. However, these two kinds of methods either require specialized hardware or sacrifice performance.

In this work, we revisit the problem of event-based fea-

ture extraction and attempt to fully utilize temporal and polarity information of events with a Transformer-based backbone, while preserving its spatial modeling capability. To this end, we first propose a novel event representation called Group Token, which groups events according to their timestamps and polarities. Then we propose Group Event Transformer (GET) to extract features from Group Tokens. We design two key modules for GET: Event Dual Self-Attention (EDSA) block and Group Token Aggregation (GTA) module. The EDSA block acts as the main self-attention module to extract correlations between different pixels, polarities, and times within Group Tokens. It maintains efficient and effective feature extraction by performing local Spatial Self-Attention and Group Self-Attention and establishing dual residual connections. GTA is placed between two stages to achieve reliable spatial and group-wise token aggregation. It achieves global communication in both the spatial and temporal- polarity domains by using a novel overlapping group convolution. Figure 1 shows the overview of a 3-stage GET network with the initial group number of 6. The grouped tokens are fused to produce larger groups, in which the features gradually gain larger spatial and temporal-polarity receptive fields, helping to better characterize objects.

These novel designs make GET achieve state-of-the-art performance on four event-based classification datasets (84.8% on Cifar10-DVS, 99.7% on N-MNIST, 96.7% on N-CARS, and 97.9% on DVS128Gesture) and two event-based object detection datasets (47.9% and 48.4% mAP on Gen1 and 1Mpx). These results all demonstrate our proposed network is beneficial for event-based feature extraction. Our contributions are summarized in the following.

- We propose a new event representation, called Group Token, that groups asynchronous events based on their timestamps and polarities.
- We devise the Event Dual Self-Attention block, enabling effective feature communication in both the spatial and temporal-polarity domains.
- We design the Group Token Aggregation module, which uses the overlapping group convolution to integrate and decouple information in both domains.
- Based on the above representation and modules, we develop a powerful Transformer backbone for event-based vision, called GET. Experimental results on both classification and object detection tasks demonstrate its superiority.

2. Related Work

Event Representation. As the event streams are asynchronous and sparse, it is necessary to convert them into appropriate alternative representations. Image-like event representations are widely utilized, such as voxel grid [76], event histogram [46] and time surface [30, 55]. However, these representations discard partial information of events, leading to a degradation in performance. There are also works introducing learning-based representations [5, 45], but these approaches introduce redundancy and latency to the network. Additionally, some recent works have accomplished graph-based representations [34, 51], but have yet to yield satisfactory performance.

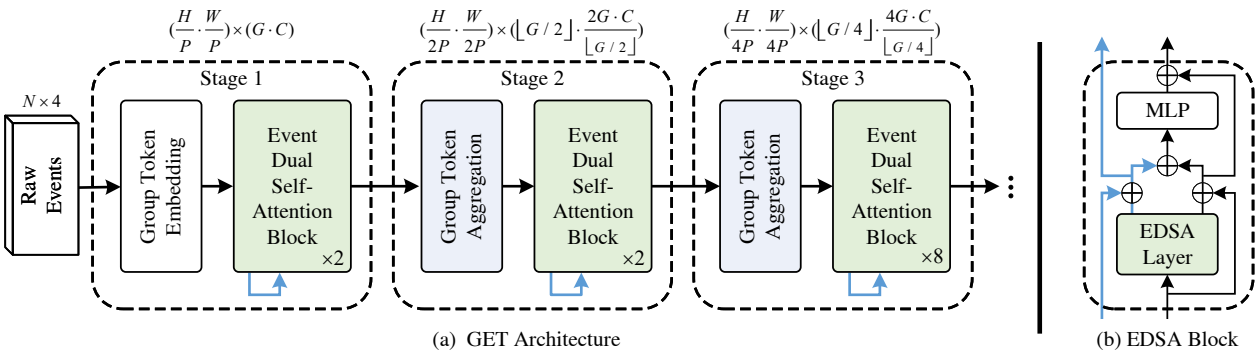
Self-Attention Mechanism. Self-attention mechanism [60, 15] is an integral part of deep neural networks, which was first widely used in the field of natural language processing (NLP). In the computer vision community, Vision Transformer (ViT) [16] is a seminal work that contributes an artful vision backbone by applying self-attention to images. Dual attention [19] introduces a novel self-attention method that extracts channel information and saves it into the spatial domain and greatly improves performance. But it is still limited by the high computational costs of global attention calculation, and its direct fusion operation of spatial and channel feature maps also results in interference. More recently, many variants of ViT [58, 69, 25, 33, 43] are proposed to improve performance and data/computation efficiency by introducing the local self-attention operation or building hierarchical Transformer architectures.

Event-Based Transformer. Transformer networks have gained popularity in various event-based tasks, such as classification [50], object detection [24], and video reconstruction [63]. Some of them employ self-attention operations on feature maps obtained through CNN-based backbones [56, 75, 63], as global attention mechanisms they adopt are computationally heavy. Recently, some studies have focused on directly extracting features from event representations [72, 62, 49, 24]. However, their Transformer designs are often unable to make full use of the provided event properties like time and polarity.

3. Method

3.1. Main Architecture

Our proposed GET is a 3-stage Transformer-based network, the architecture of which is illustrated in Figure 2(a). Three modules are utilized within stages: GTE module, EDSA block, and GTA module. At Stage 1, raw events are embedded into Group Tokens via GTE. Then the tokens are fed to 2 sequential EDSA blocks. At Stage 2&3, one GTA module is first deployed, followed by 2 and 8 EDSA blocks, respectively. The structure of an EDSA block, which combines an EDSA layer, followed by an MLP layer and normalization layers, is shown in Figure 2(b). It partitions Group Tokens and applies local self-attention in both spatial and temporal-polarity dimensions. The GTA module, which involves overlapping group convolution followed by max pooling, is employed to integrate and decouple spatial and



(a) GET Architecture (b) EDSA Block

Figure 2. (a) The architecture of a 3-stage GET; (b) The EDSA block with dual residual connections, blue arrows mark the feature maps after GSA refinement. (All figures omit the normalization and activation layers.)

temporal-polarity information between two stages. Residual connections of feature maps after Group Self-Attention refinement are marked by blue arrows, and there are no cross-stage residual connections. The output feature maps are sent to heads of different downstream tasks like classification and object detection.

3.2. Group Token Embedding

The Group Token Embedding (GTE) module is designed to convert an event stream to group tokens. The event streams from an event camera with the $H \times W$ resolution, are described as $(\vec{t}, \vec{p}, \vec{x}, \vec{y})$, where $\vec{t}$, $\vec{p}$ and $(\vec{x}, \vec{y})$ are the time, polarity and location coordinates. We first discretize the time of asynchronous events to K intervals and encode location coordinates into the rank of $P \times P$ patches and positions of patches. We also define discretized time $\vec{d_t}$, patch rank $\vec{pr}$, and patch position $\vec{pos}$ as:

$$\begin{cases} \vec{d_t} &= \lfloor K \times (\vec{t} - t_0) \,/\, (t_{end} - t_0 + 1) \rfloor \\ \vec{pr} &= \lfloor \vec{x} \mod P \rfloor + \lfloor \vec{y} \mod P \rfloor \times P \\ \vec{pos} &= \lfloor \vec{x} \,/\, P \rfloor + \lfloor \vec{y} \,/\, P \rfloor \times (W/P) \end{cases} . \quad (1)$$

Then, we map the polarity array and the above three arrays to a single 1D array, which can be described as:

$$\vec{l} = (K \cdot H \cdot W) \cdot \vec{p} + (H \cdot W) \cdot \vec{d_t} + (\frac{H \cdot W}{P^2}) \cdot \vec{pr} + \vec{pos}. \quad (2)$$

Using 1D bin count operation with weights of $\vec{1}$ and relative time $(\vec{t} - t_0)/(t_{end} - t_0)$, two 1D arrays with length $H \cdot W \cdot 2K$ are generated. After concatenation and reshape operations, we get event representations with the shape $(\frac{H}{P} \cdot \frac{W}{P}) \times (2K \cdot 2P^2)$.

Early convolutions have been proven effective for Transformer networks [66, 9]. Thus we further embed the representations with a 3×3 group convolution layer following an MLP layer. Finally, we get the Group Tokens with the dimension $(\frac{H}{P} \cdot \frac{W}{P}) \times (G \cdot C)$, where C indicates the channel number of each group. The variable G is the number of groups with different combinations of time interval and polarity. In other words, G is equal to either $2K$ or $2 \cdot \frac{K}{2}$, depending on the group division of the convolution layer.

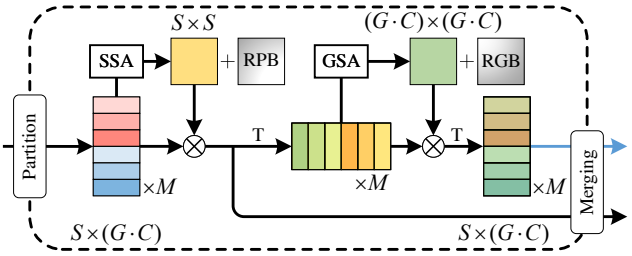


Figure 3. **Event Dual Self-Attention process.** SSA is first performed on the partitioned tokens. GSA is then applied. RPB and RGB are added to two attention maps.

3.3. Event Dual Self-Attention Block

The EDSA block is composed of an EDSA layer, an MLP layer, and normalization layers, as shown in Figure 2(b). It is designed to effectively extract features in both the spatial and temporal-polarity domains. However, as the contextual information in two domains varies, the direct fusion of two kinds of feature maps usually causes interference and biases the network to a local optimum. To address this issue, the EDSA block employs dual residual connections, as shown by black and blue arrows. These dual residual connections avoid information loss in a certain domain by retaining information from previous layers separately in both domains.

Figure 3 shows the structure of an EDSA layer that is included in an EDSA block. For efficient computation, the first step in the EDSA layer is partitioning the feature maps into M non-overlapping windows with size S to compute parallel self-attention, as in Swin-Transformer [43]. Then, Spatial Self-Attention (SSA) is performed on each window with shape $S \times (G \cdot C)$ to capture spatial contextual information. The SSA operation can be described as:

$$\begin{aligned} Q, K, V &= XW^Q, XW^K, XW^V \\ \text{SSA}(Q, K, V) &= \text{Softmax}(\frac{QK^T}{\sqrt{G \cdot C}} + B_p)V, \end{aligned} \quad (3)$$

where $W^Q, W^K, W^V \in \mathbb{R}^{(G \cdot C) \times (G \cdot C)}$ are linear weights to transform input into query Q, key K, and value V along

the channel dimension. B_p is the relative position bias (RPB), as in [52, 48]. The attention map is generated by a dot product between Q and K. The feature maps are yielded by a matrix multiplication operation between the attention map and the value V.

After performing SSA, we transpose the output feature maps into shape $(G \cdot C) \times S$ to perform group self-attention (GSA). Our proposed GSA captures temporal and polarity contextual information of the Group Tokens to refine the semantic features. The GSA operation can be described as:

$$\begin{aligned} Q_g, K_g, V_g &= X^T W_g^Q, X^T W_g^K, X^T W_g^V \\ \text{GSA}(Q_g, K_g, V_g) &= \text{Softmax}(\frac{Q_g K_g^T}{\sqrt{S}} + B_g) V_g, \end{aligned} \tag{4}$$

where $W_g^Q, W_g^K, W_g^V \in \mathbb{R}^{S \times S}$ are linear weights to transform input along the spatial dimension. We include a relative group bias (RGB) $B_g \in \mathbb{R}^{(G \cdot C) \times S}$ to guide the attention process. The indexes of relative group bias are between $-G+1$ and $G-1$, denoting different feature groups. The attention map is generated by a dot product between Q_g and K_g. A matrix multiplication operation between the attention map and V_g calculates the feature maps.

Finally, the EDSA layer outputs two feature maps: one obtained by applying SSA only (black arrow), and the other obtained by applying both SSA and GSA (blue arrow). The EDSA block guarantees effective feature communication in both the spatial and temporal-polarity domains.

3.4. Group Token Aggregation

GTA is a novel approach that we use to address the limitations of existing token aggregation methods in hierarchical Transformers. To maintain translational equivariance, well-organized token aggregation is important, as highlighted by previous works [59, 74]. For example, Swin-Transformer [43] and Nested-Transformer [74] use shift-window and conv-pool methods to integrate spatial information into the channel domain. However, these methods destroy the group correlations and are unable to integrate information in the temporal-polarity domain. In contrast, GTA utilizes a new convolution method called overlapping group convolution, which effectively integrates and decouples information in both domains.

Overlapping group convolution is designed based on the concept of group convolution [28]. As illustrated in Figure 4, the input feature maps of overlapping group convolution have shape $(\frac{H}{P} \cdot \frac{W}{P}) \times (G \cdot C)$. It is then divided into $\lfloor G/2 \rfloor$ new overlapping groups and convolved with a set of kernels that have been partitioned into the same number of groups $\lfloor G/2 \rfloor$. GTA has two parameters: The group-wise kernel (GK) denotes the input group number of each kernel and the group-wise stride (GS) denotes the group-wise distance

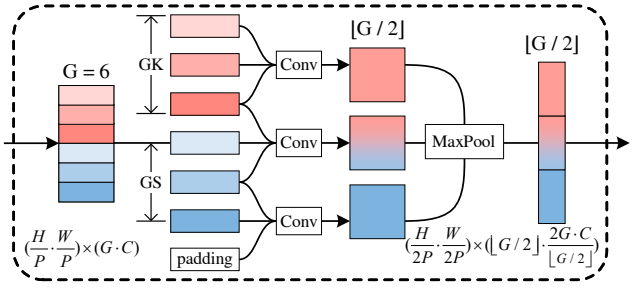


Figure 4. **Group Token Aggregation process.** We achieve reliable spatial and temporal-polarity information integration by proposing overlapping group convolution.

between two inputs. In GET, GK and GS are chosen as $min(\lfloor G/2 \rfloor + 1, 3)$ and $min(\lceil G/2 \rceil - 1, 2)$. A padding with shape $(\frac{H}{P} \cdot \frac{W}{P}) \times C$ is added if needed.

The GTA module consists of a 3×3 overlapping group convolution, followed by a normalization layer and a 3×3 max pooling operation, where 3×3 denotes the kernel size. After using overlapping group convolution, the channel number is doubled, and the group number is halved. The features in each group gain a larger $(3\times)$ receptive field in both the spatial and temporal-polarity domains. This gradual inter-group integration is similar to inter-pixel integration in CNN networks, where feature maps gain a larger receptive field as the network deepens. Moreover, the multi-kernel design of overlapping group convolution can reduce computational cost and model size without compromising performance [28]. The max pooling further downsamples the tokens. This results in a final output shape of $(\frac{H}{2P} \cdot \frac{W}{2P}) \times (\lfloor G/2 \rfloor \cdot \frac{2G \cdot C}{\lfloor G/2 \rfloor})$. Through ablative experiments, it is proved that GTA achieves reliable information integration in both the spatial and temporal-polarity domains.

4. Experiments

We first demonstrate the comparison results of GET and other state-of-the-art works on four event-based classification datasets and two event-based object detection datasets. Then we present ablative experimental results on both tasks to analyze the major contributions of the proposed representation and modules. Finally, we show the visualizations of feature maps and detection results.

4.1. Experimental Setup

Datasets. We evaluate our proposed methods on four event-based classification datasets, which are Cifar10-DVS [31], N-MNIST [29], N-CARS [55], and DVS128Gesture [1]. We also test the methods on two representative large-scale object detection datasets, which are the Gen1 dataset [10] and the 1Mpx dataset [47].

The Cifar10-DVS dataset contains 10,000 event streams converted from the up-sampled images in the Cifar10 dataset [57]. The resolution of these streams is 128×128

		Cifar10-DVS	N-MNIST	N-CARS	
Methods	Input Type	top-1 acc. (%)	top-1 acc. (%)	top-1 acc. (%)	Params (M)
HATS [55]	Frame	52.4	99.1	90.2	-
AMAE [14]	Frame	62.0	98.3	93.6	21.8
MVF-Net [12]	Frame	55.8	98.1	92.7	33.5
RG-CNNs [4]	Voxel	54.0	99.0	91.4	19.5
EV-VGCNN [13]	Voxel	65.1	99.4	95.3	0.8
VMV-GCN [67]	Voxel	66.3	99.5	93.2	0.9
DT-SNN [65]	Spike	60.5	99.5	-	-
PLIF [17]	Spike	74.8	99.6	-	4.7 - 17.1
EvS [35]	Graph	68.0	99.1	93.1	-
Swin-T v2 [42]	Token	77.2	99.3	92.8	6.9
Nested-T [74]	Token	78.1	99.3	93.3	4.2
Ours	Token	**84.8**	**99.7**	**96.7**	4.5

Table 1. Classification performance on the Cifar10-DVS, N-MNIST, and N-CARS datasets.

and there are 10 different classes. The N-MNIST dataset includes 70,000 event streams converted from the original MNIST dataset [11], which have 10 different classes and a resolution of 34×34. The N-CARS dataset includes 12,336 car samples and 11,693 non-car samples. The resolutions of samples in N-CARS are different. The highest resolution is not higher than 128×128. The DVS128Gesture dataset contains 1,342 real-world event streams collected by a 128×128 event camera. There are 11 different gesture classes (including a class of random movements).

The Gen1 dataset is the most commonly used event-based object detection dataset, which consists of 39 hours of events. The Gen1 dataset is collected with a resolution of 304×240 and contains 2 object classes. The labeling frequency of the Gen1 dataset is 20 Hz. Another dataset, the 1Mpx dataset contains 14.65 hours of events. The resolution of samples in the 1Mpx dataset is 1280×720. The labeling frequency of this dataset is 60 Hz, and the number of bounding boxes exceeds 25M. There are 7 labeled object classes. Following [32, 47], we only utilize 3 classes (car, pedestrian, and two-wheeler) for performance comparison.

Implementation Details. For classification experiments, we follow the official train/test split for N-MNIST, N-CARS, and DVS128Gesture datasets for fair comparisons. For the Cifar10-DVS dataset, we follow the split adopted by previous studies [65, 17] to select samples for testing. Our approach applies random horizontal flip and random spatial-temporal crop augmentations on raw events to classification tasks. Since the DVS128Gesture dataset includes both right and left-hand movements, target transformation is applied after using a horizontal flip. Random horizontal flip augmentation is not used on the N-MNIST dataset. We also include Mixup [71] and Cutmix [70] to overcome the over-fitting problem on certain datasets. We train our classification models from scratch for 1000 epochs.

		DVS128Gesture	
Methods	Input Type	top-1 acc. (%)	Params (M)
PLIF [17]	Spike	97.6	1.7
SLAYER [54]	Spike	93.6	-
TORE [3]	Tore	96.2	5.9
PointNet++ [61]	Clouds	95.3	3.5
EvT [50]	Token	96.2	0.5
Swin-T v2 [42]	Token	93.2	7.1
Nested-T [74]	Token	93.8	4.2
Ours	Token	**97.9**	4.5

Table 2. Classification performance on the DVS128Gesture dataset, which includes long time-range samples.

For object detection experiments, the train/test split of Gen1 and 1Mpx datasets are predefined. We choose the most commonly used 50 ms as the time interval of each sample and follow the dataset processing of previous works [24, 47, 32] to remove small bounding boxes and downsample events. We follow the same training strategy as RVT [24]. Our detection models are trained from scratch for 400000 steps.

We use a 3-stage GET with an embedding dimension of 48 on classification tasks. The group number G of Cifar10-DVS, N-MNIST, N-CARS, and DVS128Gesture datasets are chosen as 12, 12, 12, and 24, respectively. For object detection tasks, we integrate a 4-stage GET with an embedding dimension of 72 into RVT [24]. The G chosen for Gen1 and 1Mpx datasets is 12. We utilize 8 NVIDIA Tesla A800 GPUs for training. When comparing the running time, we only use one NVIDIA GTX 1080Ti GPU.

4.2. Experimental Results

Classification. We compare GET with other state-of-the-art methods on Cifar10-DVS, N-MNIST, N-CARS, and

DVS128Gesture datasets and report the results in Table 1 and Table 2. Swin-Transformer v2 (Swin-T v2) [42] and Nested-Transformer (Nested-T) utilizing patched voxel grid [76] as input are also included in the comparison. The embedding dimension of them is also chosen as 48. The time bin number of the voxel grid they used ranges from 4 to 48, but we only list their optimal results. We utilize top-1 accuracy (top-1 acc.) to evaluate the classification performance. We also provide parameter numbers (Params) of the models for a fair comparison.

On the Cifar10-DVS dataset, GET outperforms other methods by a large margin (84.8% vs. 74.8%, PLIF). Compared with the Nested-T, its top-1 accuracy also increases by 6.7%. On the N-MNIST dataset, GET reaches the highest performance (99.7% vs. 99.6%, PLIF). Compared with Nested-T, its top-1 accuracy is increased by 0.4%. On the N-CARS dataset, there is a significant performance gain using our method (96.7% vs. 95.3%, EV-VGCNN). Its top-1 accuracy is increased by 3.4% compared with Nested-T.

On the DVS128Gesture dataset, which contains long-time-range event streams. SNN-based methods [68, 17, 27, 54] split an event stream into T bins and feed them iteratively into the network. The inference process is executed T times. As a result, many works [3, 50] refrain to compare with the SNN-based works on this dataset. However, the encouraging results in Table 2 indicate that GET achieves better performance (97.9% vs. 97.6%, PLIF) with only a single-pass inference. This means that our methods can better learn the long-range temporal information of event samples without introducing heavy 3D / recurrent designs.

Object Detection. We further evaluate GET on two large-scale event-based object detection datasets, which are the Gen1 dataset and the 1Mpx dataset. The results compared with state-of-the-art works are reported in Table 3 and Table 4. We use the COCO mean average precision (mAP) [37] as the main metric. We also provide the comparison on running time (runtime), which includes the event conversion time and the network inference time.

The labeled bounding boxes of the Gen1 dataset and the 1Mpx dataset are generated by detection on frames, thus many areas with few events may still be labeled, since event cameras only capture moving objects and objects with illumination changes. Therefore, some works use LSTM or other memory mechanisms to enhance performance [24, 47, 32, 41]. To comprehensively evaluate the capability of the networks, we compare the performance in two scenarios. The first scenario is that the comparison networks don't utilize memory modules for enhancement. The second scenario is the opposite. For GET, we use the YOLOX framework [22] and place the ConvLSTM layers [53] at the backbone when comparing with the memory-enhanced networks as in RVT [24]. The comparison results for object detection are shown in Table 3 and Table 4.

	Gen1		
Methods	mAP (%)	runtime (ms)	Params (M)
AEGNN [51]	16.3	-	-
SNN-SSD [7]	18.9	-	8.2
SAM [36]	35.5	-	-
FS [8]	39.6*	41.2	-
RED [47]	- / 40.0	- / 16.7	24.1
ASTMNet [32]	38.2 / 46.7	28.6 / 35.6	39.6
RVT-B [24]	32.0 / 47.2	- / 10.2	18.5
Swin-T v2 [42]	34.3 / 45.5	25.2⋆ / 26.6⋆	17.6 / 21.1
Nested-T [74]	35.1 / 46.3	24.8⋆ / 25.9⋆	18.7 / 22.2
Ours	**38.7** / **47.9**	15.9⋆ / 16.8⋆	18.4 / 21.9

Table 3. Detection performance on the Gen1 dataset. blue: The memory-enhanced results. ∗: The result is in COCO AP50. ⋆: The runtime includes event conversion time and inference time.

	1Mpx		
Methods	mAP (%)	runtime (ms)	Params (M)
SAM [36]	23.9	-	-
UDA [44]	48.0*	-	-
RED [47]	29.0 / 43.0	- / 39.3	24.1
ASTMNet [32]	40.3 / 48.3	59.8 / 72.3	>39.6
RVT-B [24]	- / 47.4	- / 11.9	18.5
Swin-T v2 [42]	36.0 / 46.4	33.6⋆ / 34.5⋆	17.6 / 21.1
Nested-T [74]	37.5 / 46.0	32.1⋆ / 33.5⋆	18.7 / 22.2
Ours	**40.6** / **48.4**	17.1⋆ / 18.2⋆	18.4 / 21.9

Table 4. Detection performance on the 1Mpx dataset. blue: The memory-enhanced results. ∗: The result is in COCO AP50. ⋆: The runtime includes event conversion time and inference time.

Table 3 shows that GET performs best on the Gen1 dataset. The state-of-the-art network without using any memory mechanism is ASTMNet, with an mAP of 38.2%. GET performs better at 38.7%, while the parameter number is less than its 47%. It also maintains an optimal mAP of 47.9% when using the memory mechanism, which is 0.7% larger than the RVT-B of 47.2%. Compared with Nested-T (with YOLOX framework, using patched voxel grid as event representation), the gains with and without memory mechanism are 3.6% and 1.6% respectively. GET also achieves the optimal result on the 1Mpx dataset, as shown in Table 4. When the memory mechanism is not used, GET outperforms the state-of-the-art network ASTMNet (40.6% vs. 40.3%). With the help of ConvLSTM layers, GET achieves the highest mAP of 48.4%, compared with ASTMNet of 48.3%. Compared with Nested-T, the gains with and without memory mechanism are 3.1% and 2.4%.

On both Gen1 and 1Mpx datasets, GET is the fastest end-to-end method. Its combined runtime is even smaller than the data preprocessing time of other methods.

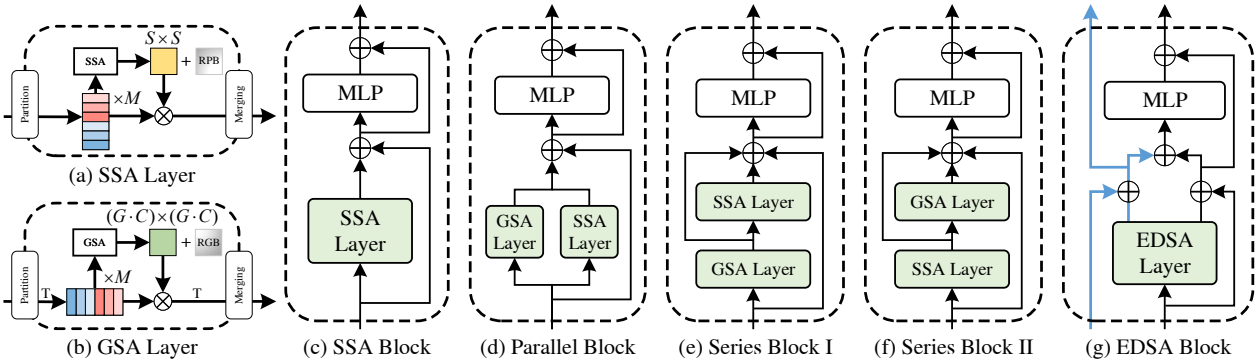


Figure 5. (a) The SSA layer. (b) The GSA layer. (c) The attention block with SSA operation only. (d) The attention block with both SSA and GSA operations, which are implemented in parallel. (e) The serial attention block performs the GSA operation first and then the SSA operation. (f) The serial attention block performs the SSA operation first and then the GSA operation. (g) The EDSA block contains dual residual connections.

				CIFAR10-DVS		1Mpx	
Models	GTE	EDSA Block	GTA	top-1 acc. (%)	Param. (M)	mAP (%)	runtime (ms)
Model I	×	×	×	77.2	4.2	37.5	32.1
Model II	✓	×	×	79.9 (+2.7)	4.4	39.1 (+1.6)	16.2
Model III	✓	✓	×	81.5 (+4.3)	4.7	40.0 (+2.5)	18.2
Model IV	✓	×	✓	79.4 (+2.2)	4.2	39.3 (+1.8)	14.9
GET	✓	✓	✓	**84.8** (+7.6)	4.5	**40.6** (+3.1)	17.1

Table 5. Performance of GET on the CIFAR10-DVS and 1Mpx datasets with different modules. ×: modules of Nested-T is used.

Representations	Group Token	Event Histogram [46]	Voxel Grid [76]	Time Surface [30]	TORE [3]
top-1 acc. (%)	**84.8**	75.8	78.4	80.2	80.4
Time (s / 10^8evs)	**0.052**	0.374	0.390	4.235	16.027

Table 6. Classification performance on the Cifar10-DVS dataset, and the time cost of converting 10^8 events into various representations.

	CIFAR10-DVS	
Methods	top-1 acc. (%)	Param. (M)
SSA Block	79.4	4.2
Parallel Block	78.5	4.5
Series Block I	75.0	4.5
Series Block II	81.9	4.5
EDSA Block	**84.8**	4.5

Table 7. Classification performance of GET on the CIFAR10-DVS dataset with different self-attention blocks.

4.3. Ablation Studies

To analyze the proposed methods for GET, we conduct a series of ablative experiments on the CIFAR10-DVS, DVS128Gesture, and 1Mpx datasets. We construct four variants Model I-IV, which have different combinations of proposed modules. Table 5 reports the performance of these variants and GET.

Group Token Embedding. Group Token enriches patch temporal-polarity information, yielding a 2.7% increase in top-1 accuracy and a 1.6% rise in mAP. Furthermore, the implementation of this approach has resulted in a 51% reduction in runtime. Hence, Group Token is an efficient and effective method to improve performance by enhancing temporal-polarity information.

We also conduct experiments to convert 10^8 events to different representations. As shown in Group Token Embedding 6, our scheme only requires only 0.052s, which is much less than using other methods' officially released codes. For performance comparison, we use VIT's [16] patch embedding module for tokenizing other representations. It can be observed that our proposed Group Token outperforms others by a large margin (84.8% vs. 80.4%).

Event Dual Self-Attention. EDSA is an effective approach to extract features from both the spatial and temporal-polarity domains while maintaining cross-pixel and cross-group communications. In comparison with the commonly used SSA approach, EDSA improves the top-1 accuracy and mAP by 1.6% and 0.9%, respectively. However, there is a slight increase in the number of parameters and running time. To demonstrate the superiority of EDSA's block structure, we compared it with other possible block structures, as illustrated in Figure 5. Table 7 presents the performance

	CIFAR10-DVS		
Modules / Models	Swin-T v2 [42]	Nested-T [74]	GET
Token Embedding	78.4	78.4	**84.8**
Self-Attention Block	79.4	79.8	**84.8**
Token Aggregation	80.4	81.0	**84.8**

Table 8. Module substitution comparison on the CIFAR10-DVS dataset. The number indicates the performance of a GET variant with a specific module replaced by the corresponding module of Swin-T v2 and Nested-T.

Datasets / #Group	4	6	12	24	48
Cifar10-DVS	79.9	81.1	**84.8**	82.3	77.6
DVS128Gesture	94.8	95.8	97.2	**97.9**	96.9

Table 9. Classification performance of GET on the Cifar10-DVS and DVS128Gesture datasets with different group numbers.

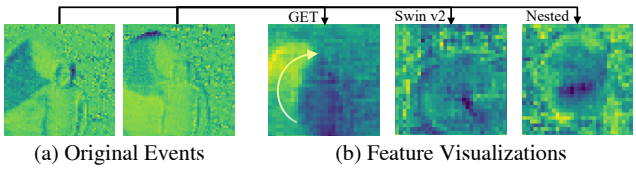
of GET using these attention blocks, indicating that EDSA block with dual residual connections achieves the best result (84.8% vs. 81.9%). Therefore, EDSA is a promising approach for feature extraction in computer vision tasks.

Group Token Aggregation. GTA leverages overlapping group convolution to decouple spatial and temporal-polarity information at an earlier stage. The resulting model is more parameter-efficient, with parameters reduced by 0.2 M, and faster, with a 1.3 ms reduction in running time. However, without EDSA to compensate for the reduced inter-channel interaction caused by group convolution, GTA will lead to a slight decrease in performance.

GET. When combined with the GTE module, EDSA block, and GTA module, GET achieves the highest top-1 accuracy and mAP of 84.8% and 40.6%.

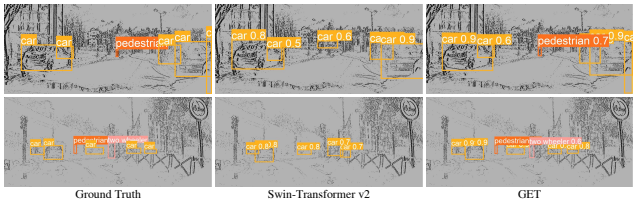
Module Superiority. To demonstrate the superiority of our proposed modules, we compare them with corresponding modules from other networks. The results in Table 8, show that our modules outperform the rest on the CIFAR10-DVS dataset. Specifically, the token embedding modules of Swin-T v2 and Nested-T both embed voxel grids into patches and achieve an accuracy of 78.4%. The self-attention modules of the two models (SSA Block in Figure 5, w/ and w/o shift-window operation) result in 79.4% and 79.8%. And the token aggregation modules of Swin-T v2 and Nested-T achieve 80.4% and 81.0% accuracy by using patch merging and conv-pool methods. Our proposed modules outperform all with an accuracy of 84.8%.

Group Number. Table 9 provides the performance on the Cifar10-DVS and DVS128Gesture datasets when choosing different group numbers G. The Cifar10-DVS dataset contains short-time-range event streams. As a result, when G is chosen as 12 ($2K = 24$), GET accurately classifies most objects and results in an 84.8% top-1 accuracy. As G increases, performance deteriorates because events in each group become sparse. The DVS128Gesture dataset contains long-time-range event streams. If G is small, the action overlaps within a group, degrading performance. When G is chosen as 24 ($2K = 48$), we get the highest accuracy of 97.9%.



(a) Original Events (b) Feature Visualizations

Figure 6. (a) Visualizations of original events (right-hand counterclockwise waving in DVS128Gesture dataset). (b) Feature visualizations of different Transformers utilizing EDSA and SSA.



Ground Truth Swin-Transformer v2 GET

Figure 7. Visualizations of the detection results on the 1Mpx dataset. Compared with the Swin-T v2, GET detects most objects, while the bounding boxes are more accurate.

4.4. Visualizations

we present visualizations of the feature maps extracted by GET and other Transformers in Figure 6. It can be observed that only GET, through the utilization of EDSA, successfully captures counterclockwise motion. This becomes evident as we compare the results. The other two classifiers that use SSA as the token mixer inaccurately categorized the sample as right-hand clockwise waving.

We also present object detection results on the 1Mpx dataset in Figure 7. It can be seen that GET detects the most objects compared with Swin-T v2, especially when objects are intertwined with the background. The bounding boxes detected by GET are also more accurate.

5. Conclusion

In this paper, we propose a novel event-based vision backbone, called Group Event Transformer (GET), that effectively utilizes the temporal and polarity information of events while preserving the fair spatial modeling capability of traditional Transformer-based backbones. The proposed backbone incorporates the Group Token representation, the EDSA block, and the GTA module, which enable effective feature communication and integration in both the spatial and temporal-polarity domains. The experimental results on four event-based classification datasets (Cifar10-DVS, N-MNIST, N-CARS, and DVS128Gesture) and two event-based object detection datasets (1Mpx and Gen1) all show that our method achieves superior performance over state-of-the-art methods.

6. Acknowledgement

This work is in part supported by the National Key R&D Program of China under Grant 2020AAA0108600 and the National Natural Science Foundation of China under Grant 62032006.

References

[1] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

[2] Anastasios N. Angelopoulos, Julien N.P. Martel, Amit P. Kohli, Jörg Conradt, and Gordon Wetzstein. Event-based near-eye gaze tracking beyond 10,000 hz. *IEEE Transactions on Visualization and Computer Graphics*, 2021. 1

[3] Raymond Baldwin, Ruixu Liu, Mohammed Mutlaq Almatrafi, Vijayan K Asari, and Keigo Hirakawa. Time-ordered recent event (tore) volumes for event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence (Early Access)*, 2022. 5, 6, 7

[4] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-based spatio-temporal feature learning for neuromorphic vision sensing. *IEEE Transactions on Image Processing*, 2020. 5

[5] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. A differentiable recurrent surface for asynchronous event-based data. In *The European Conference on Computer Vision (ECCV)*, 2020. 2

[6] Loïc Cordone, Benoît Miramond, and Philippe Thierion. Object detection with spiking neural networks on automotive event data. In *The International Joint Conference on Neural Networks (IJCNN)*, 2022. 1

[7] Loic Cordone, Benoît Miramond, and Phillipe Thierion. Object detection with spiking neural networks on automotive event data. In *The IEEE International Joint Conference on Neural Networks (IJCNN)*, July 2022. 6

[8] Brian Crafton, Andrew Paredes, Evan Gebhardt, and Arijit Raychowdhury. Hardware-algorithm co-design enabling efficient event-based object detection. In *The IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2021. 6

[9] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3

[10] Pierre de Tournemire, Davide Nitti, Etienne Perot, Davide Migliore, and Amos. Sironi. A large scale event-based detection dataset for automotive. *arXiv preprint arXiv:2001.08499*, 2020. 4

[11] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 2012. 5

[12] Yongjian Deng, Hao Chen, and Youfu Li. Mvf-net: A multi-view fusion network for event-based object classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022. 5

[13] Yongjian Deng, Hao Chen, Hai Liu, and Youfu Li. A voxel graph cnn for object classification with event cameras. *arXiv preprint arXiv:2106.00216*, 2021. 5

[14] Yongjian Deng, Youfu Li, and Hao Chen. Amae: Adaptive motion-agnostic encoder for event-based object classification. *IEEE Robotics and Automation Letters*, 2020. 5

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *The Conference of the North American Chapter of the Association for Computational*, 2019. 2

[16] 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. *International Conference on Learning Representations (ICLR)*, 2021. 1, 2, 7

[17] 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 *The IEEE International Conference on Computer Vision (ICCV)*, 2021. 5, 6

[18] Tobias Fischer and Michael Milford. Event-based visual place recognition with ensembles of temporal windows. *IEEE Robotics and Automation Letters*, 2020. 1

[19] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[20] Guillermo Gallego, Tobi Delbrück, and Garrick Orchard, et al. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1

[21] Shan Gao, Guangqian Guo, and C. L. Philip Chen. Event-based incremental broad learning system for object classification. In *The IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2019. 1

[22] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 6

[23] Daniel Gehrig, Antonio Loquercio, Konstantinos G. Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 1

[24] Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras. *arXiv preprint arXiv: 2212.05598*, 2022. 1, 2, 5, 6

[25] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv: 2103.00112*, 2021. 1, 2

[26] Xueyan Huang, Yueyi Zhang, and Zhiwei Xiong. Progressive spatio-temporal alignment for efficient event-based mo-

tion estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[27] Simone Undri Innocenti, Federico Becattini, Federico Pernici, and Alberto Del Bimbo. Temporal binary representation for event-based action recognition. In *The International Conference on Pattern Recognition (ICPR)*, 2021. 1, 6

[28] Y. Ioannou, D. Robertson, R. Cipolla, and A. Criminisi. Deep roots: Improving cnn efficiency with hierarchical filter groups. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

[29] Laxmi R. Iyer, Yansong Chua, and Haizhou Li. Is neuromorphic mnist neuromorphic? analyzing the discriminative power of neuromorphic datasets in the time domain. *Frontiers in Neuroscience*, 2021. 4

[30] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E. Shi, and Ryad B. Benosman. Hots: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 1, 2, 7

[31] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: An event-stream dataset for object classification. *Frontiers in Neuroscience*, 2017. 4

[32] Jianing Li, Jia Li, Lin Zhu, Xijie Xiang, Tiejun Huang, and Yonghong Tian. Asynchronous spatio-temporal memory network for continuous event-based object detection. *IEEE Transactions on Image Processing*, 2022. 5, 6

[33] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. 1, 2

[34] Yijin Li, Han Zhou, Bangbang Yang, Ye Zhang, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Graph-based asynchronous event processing for rapid object recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[35] Yijin Li, Han Zhou, Bangbang Yang, Ye Zhang, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Graph-based asynchronous event processing for rapid object recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, 2021. 5

[36] Zichen Liang, Guang Chen, Zhijun Li, Peigen Liu, and Alois Knoll. Event-based object detection with lightweight spatial attention mechanism. In *The IEEE International Conference on Advanced Robotics and Mechatronics (ICARM)*, 2021. 6

[37] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *The European Conference on Computer Vision (ECCV)*, 2014. 6

[38] Mengyun Liu, Na Qi, Yunhui Shi, and Baocai Yin. An attention fusion network for event-based vehicle object detection. In *The IEEE International Conference on Image Processing (ICIP)*, 2021. 1

[39] Qianhui Liu, Dong Xing, Huajin Tang, De Ma, and Gang Pan. Event-based action recognition using motion information and spiking neural networks. In *The International Joint Conference on Artificial Intelligence (IJCAI-21)*, 2021. 1

[40] Qianhui Liu, Dong Xing, Huajin Tang, De Ma, and Gang Pan. Event-based action recognition using motion information and spiking neural networks. In *The International Joint Conference on Artificial Intelligence (IJCAI)*, 2021. 1

[41] Xu Liu, Jianing Li, Xiaopeng Fan, and Yonghong Tian. Event-based monocular dense depth estimation with recurrent transformers. *arXiv preprint arXiv:2212.02791*, 2022. 6

[42] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5, 6, 8

[43] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *The IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 3, 4

[44] Nico Messikommer, Daniel Gehrig, Mathias Gehrig, and Davide Scaramuzza. Bridging the gap between events and frames through unsupervised domain adaptation. *IEEE Robotics and Automation Letters*, 2022. 6

[45] Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-based asynchronous sparse convolutional networks. In *The European Conference on Computer Vision (ECCV)*, 2020. 2

[46] Diederik Paul Moeys, Federico Corradi, Emmett Kerr, Philip Vance, Gautham Das, Daniel Neil, Dermot Kerr, and Tobi Delbrück. Steering a predator robot using a mixed frame/event-driven convolutional neural network. In *The International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, 2016. 1, 2, 7

[47] Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. Learning to detect objects with a 1 megapixel event camera. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 4, 5, 6

[48] Adam Roberts, Colin Raffel, Katherine Lee, Michael Matena, Noam Shazeer, Peter J. Liu, Sharan Narang, Wei Li, and Yanqi Zhou. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learn- ing Research*, 2020. 4

[49] Alberto Sabater, Luis Montesano, and Ana C. Murillo. Event transformer. a sparse-aware solution for efficient event data processing. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022. 1, 2

[50] Alberto Sabater, Luis Montesano, and Ana C. Murillo. Event transformer. a sparse-aware solution for efficient event data processing. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022. 2, 5, 6

[51] Simon Schaefer, Daniel Gehrig, and Davide Scaramuzza. Aegnn: Asynchronous event-based graph neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 6

[52] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv: 1803.02155*, 2018. 4

[53] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional lstm

network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 6

[54] Sumit Bam Shrestha and Garrick Orchard. Slayer: Spike layer error reassignment in time. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 5, 6

[55] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad B. Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 4, 5

[56] Yi Tian and J. Andrade-Cetto. Event transformer flownet for optical flow estimation. In *British Machine Vision Conference*, 2022. 1, 2

[57] Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008. 4

[58] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *The International Conference on Machine Learning (ICML)*, 2021. 2

[59] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens. Scaling local self-attention for parameter efficient visual backbones. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2

[61] Qinyi Wang, Yexin Zhang, Junsong Yuan, and Yilong Lu. Space-time event clouds for gesture recognition: From rgb cameras to event cameras. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. 5

[62] Zuowen Wang, Yuhuang Hu, and Shih-Chii Liu. Exploiting spatial sparsity for event cameras with visual transformers. In *The IEEE International Conference on Image Processing (ICIP)*, 2022. 1, 2

[63] Wenming Weng, Yueyi Zhang, and Zhiwei Xiong. Event-based video reconstruction using transformer. *The IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[64] Hao Wu, Yueyi Zhang, Wenming Weng, Yongting Zhang, Zhiwei Xiong, Zhengjun Zha, Xiaoyan Sun, and Feng Wu. Training spiking neural networks with accumulated spiking flow. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 1

[65] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019. 5

[66] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollar, and Ross Girshick. Early convolutions help transformers see better. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3

[67] Bochen Xie, Yongjian Deng, Zhanpeng Shao, Hai Liu, and Youfu Li. Vmv-gcn: Volumetric multi-view based graph cnn for event stream classification. *IEEE Robotics and Automation Letters*, 2022. 5

[68] Che-Hang Yu, Zheming Gu, Da Li, Gaoang Wang, Aili Wang, and Erping Li. Stsc-snn: Spatio-temporal synaptic connection with temporal convolution and attention for spiking neural networks. *Frontiers in Neuroscience*, 2022. 6

[69] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis E.H. Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *The IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[70] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019. 5

[71] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations (ICLR)*, 2018. 5

[72] Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, and Xin Yang. Spiking transformers for event-based single object tracking. In *The IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2

[73] Jiqing Zhang, Xin Yang, Yingkai Fu, Xiaopeng Wei, Baocai Yin, and Bo Dong. Object tracking by jointly exploiting frame and event domain. In *The IEEE International Conference on Computer Vision (ICCV)*, 2021. 1

[74] Zizhao Zhang, Han Zhang, Long Zhao, Ting Chen, , Sercan Ö. Arık, and Tomas Pfister. Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 4, 5, 6, 8

[75] Junwei Zhao, Shiliang Zhang, and Tiejun Huang. Transformer-based domain adaptation for event data classification. *The IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022. 1, 2

[76] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth and egomotion. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. 1, 2, 6, 7