EGSST: Event-based Graph Spatiotemporal Sensitive Transformer for Object Detection

Sheng Wu[1] **Hang Sheng**[1] **Hui Feng**[1,2] * **Bo Hu**[1,2]
[1] School of Information Science and Technology, Fudan University
[2] State Key Laboratory of Integrated Chips and Systems, Fudan University

Abstract

Event cameras provide exceptionally high temporal resolution in dynamic vision systems due to their unique event-driven mechanism. However, the sparse and asynchronous nature of event data makes frame-based visual processing methods inappropriate. This study proposes a novel framework, Event-based Graph Spatiotemporal Sensitive Transformer (EGSST), for the exploitation of spatial and temporal properties of event data. Firstly, a well-designed graph structure is employed to model event data, which not only preserves the original temporal data but also captures spatial details. Furthermore, inspired by the phenomenon that human eyes pay more attention to objects that produce significant dynamic changes, we design a Spatiotemporal Sensitivity Module (SSM) and an adaptive Temporal Activation Controller (TAC). Through these two modules, our framework can mimic the response of the human eyes in dynamic environments by selectively activating the temporal attention mechanism based on the relative dynamics of event data, thereby effectively conserving computational resources. In addition, the integration of a lightweight, multi-scale Linear Vision Transformer (LViT) markedly enhances processing efficiency. Our research proposes a fully event-driven approach, effectively exploiting the temporal precision of event data and optimising the allocation of computational resources by intelligently distinguishing the dynamics within the event data. The framework provides a lightweight, fast, accurate, and fully event-based solution for object detection tasks in complex dynamic environments, demonstrating significant practicality and potential for application. The source code can be found at: EGSST.

1 Introduction

There is an increasing demand for devices capable of accurately capturing targets in high-speed dynamic scenes, such as autonomous driving, where traditional CMOS or CCD visual sensors often encounter motion blur and overexposure [1, 2, 3]. In response, event cameras, which employ a novel event-driven mechanism, have garnered significant attention. Each pixel in event cameras operates independently, activating only upon detecting a brightness change. This mechanism results in a series of high-speed, asynchronous event streams with exceptionally high temporal resolution, which enables the rapid and precise capture of data [4, 5].

Our objective in processing the output data from event cameras is to effectively utilize their high temporal precision in detection tasks without inducing significant delays. However, the data format of event cameras is entirely different from that of frame-based cameras, rendering existing object detection methods based on RGB images inapplicable [6]. Some methods adopt conversion strategies, utilizing techniques such as temporal slicing to transform sparse data into dense formats, with

*Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

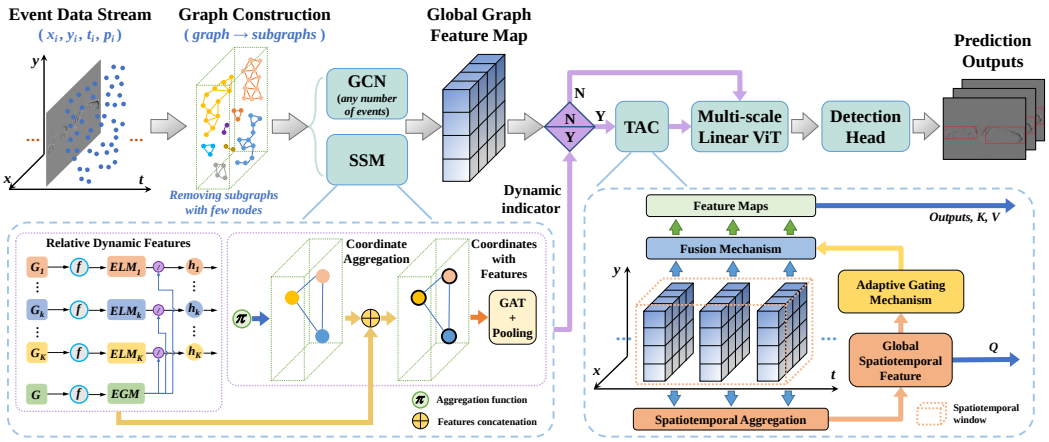

Figure 1: **An overview of the proposed EGSST framework.** The EGSST is an event-based, lightweight, and efficient framework designed for rapid object detection in event data. A graph is constructed from the data and divided into K connected subgraphs. These subgraphs are fed into a Graph Convolutional Network (GCN) [22] and a SSM. The GCN processes the subgraphs that are not removed to produce a global Graph Feature Map, which preserves both spatial and temporal information. The SSM assesses the dynamics of the entire graph and outputs a dynamic feature indicator, which includes the dynamics of each subgraph and the aggregated dynamics obtained through a Graph Attention Network (GAT) [23]. The TAC is activated based on the output from the SSM to enhance focus on the temporal dimension or to feed the graph feature maps directly into a Multi-scale Linear ViT [24]. Finally, a detection head, such as RT-DETR [25] or YOLOX [26], is employed to generate prediction outputs.

the aim of leveraging architectures like Convolutional Neural Networks (CNNs) or Transformers [2, 7, 8, 9, 10, 11]. Unfortunately, methods employing such conversion strategies often result in a loss of temporal precision. Another direct processing strategies involve methods such as Graph Neural Networks (GNNs) [12, 13, 14, 15] or Spiking Neural Networks (SNNs) [16, 17, 18, 19, 20, 21], which analyze the event stream directly, thereby preserving spatiotemporal structure and accuracy. However, the direct processing strategies often suffer from computational efficiency. The latest hybrid strategies strive to combine the strengths of previous methods, enhancing performance while maximizing retention of temporal precision. Although numerous existing strategies have made significant progress in the extraction of event data features, challenges remain in effectively managing data across spatial and temporal dimensions [12, 14].

Existing methods may result in the inefficient use of computational resources when processing data across both spatial and temporal dimensions. Because current models either apply a single algorithm to both dimensions simultaneously [12, 20] or use different algorithms for each dimension and then combine the results [27]. Importantly, processing in the temporal dimension typically consumes significantly more computational resources than processing in the spatial dimension. Consequently, the continuous processing of temporal data is likely to result in the inefficient utilization of computational resources, especially for slower-moving objects where spatial resolution may already be sufficient.

In order to address the aforementioned issue of inefficiency, we have drawn inspiration from the dynamic perception of human eyes. The human visual system naturally prioritizes rapidly moving objects within the visual field while de-prioritizing slower ones—an adaptive feature that enhances responsiveness and efficiency in dynamic environments. Ideally, artificial models can emulate this trait by preferentially processing faster-moving objects in the temporal dimension, especially when managing large volumes of continuous spatiotemporal data. Therefore, we try to develop a novel algorithm that is closer to the selective attention mechanism of the human eye, which could potentially enhance both the efficiency and effectiveness of event-based vision systems.

We propose a novel spatiotemporal fusion graph transformer framework, designed to fully leverage the powerful capability of graph to process unstructured data while also reducing unnecessary computations in handling spatio-temporal data. Firstly, the framework employs a graph structure to

model event data, thereby maintaining the original temporal fidelity and capturing crucial spatial details. A key aspect of our framework is the SSM module, which leverages the graph's ability to process unstructured data and aggregates features to effectively discern the relative dynamics of objects. This differentiation provides a critical basis for determining whether to continue processing in the temporal dimension. Subsequently, the adaptive TAC is introduced, which dynamically adjusts its activation based on the insights from the SSM. The TAC is designed to enhance the processing of highly dynamic events while reducing resource use in scenarios with low dynamics. Furthermore, the integration of a lightweight multi-scale LViT markedly enhances the processing efficiency of our system. The principal contributions of our research are as follows:

- Our model introduces a novel and efficient graph processing method by pioneering the use of connected subgraphs in the context of event cameras. This technique not only preserves the temporal data completely but also enhances the effective and precise focus on targets within the event data.
- We introduce SSM and TAC to mimic the human eye's perception of dynamics and integrate them into the Graph Transformer framework for efficient object detection in event data.
- Our model integrates Graph and Transformer technologies to enhance object detection tasks in a fully event-based manner. It is designed to be lightweight, fast, and precise, representing a novel approach that leverages the strengths of both technologies for improved performance.

2 Related Work

The unique event-driven mechanism of event cameras offers considerable potential for high-speed motion detection in dynamic environments. However, the inherent sparsity and unstructured nature of the data generated by these cameras pose significant challenges to conventional image processing techniques [4, 28]. In response, researchers have transitioned from a singular processing strategy to a hybrid approach, integrating traditional and innovative methods.

Conversion Processing Strategy involves transforming event data into dense frame-based data, making it compatible with conventional visual processing algorithms. Systems like the E2VID [29] utilize CNNs to convert asynchronous event data into video frames through time slicing. Other approaches generate continuous optical flow and intensity estimation images from event streams [30], or incorporate attention mechanisms with recurrent and convolutional networks to enhance spatio-temporal feature extraction [31]. These strategies extend the applicability of event camera data but often at the cost of reducing its high temporal resolution [32, 33, 34].

Direct Processing Strategy focuses on preserving the original asynchronous characteristics of event data to maintain its sparsity and high temporal precision. This includes direct modeling of the event stream for real-time object tracking [35] and employing spiking neural networks (SNNs) for gesture recognition [7]. Additionally, models like AEGNN [12] leverage GNNs and efficient updating strategies to retain asynchronous temporal features. While these methods effectively utilize the unique properties of event data, they face challenges in handling large volumes of sparse data, highlighting the need for further innovations in architecture [36, 20].

Hybrid Processing Strategy explores a combination of conversion and direct processing methods, or the integration of multiple model frameworks [37, 38, 39, 40, 9, 27]. For example, merging GNNs with CNNs, and integrating Transformers with recurrent neural networks, effectively captures spatio-temporal features from asynchronous events while maintaining high resolution and enhancing adaptability [38, 39]. Methods like MatrixLSTM [40] and RED [9] increase data processing efficiency by jointly extracting features. Building on these studies, a recent work, RVT [27], integrates Transformers with recurrent neural networks, significantly improving processing efficiency and accuracy. However, current models, despite their commendable performance and efficiency, cannot effectively differentiate dynamic data, leading to unnecessary computations in the time dimension.

Our research lies within the hybrid processing strategy methods. The proposed framework leverages graph capabilities to manage irregular data and incorporates a temporal attention mechanism responsive to data dynamics. It also integrates a lightweight linear visual Transformer to extract temporal and spatial information from event data. This event-based framework is designed to provide a fast, efficient, and accurate object detection solution, specifically tailored to event cameras.

3 Methodology

The proposed framework utilizes the Graph Transformer to extract graph-based features from unstructured event data and integrates both spatial and temporal attention mechanisms. As illustrated in Figure 1, the architecture and key components are depicted, providing a visual overview of how these elements are integrated. This section will outline the critical steps involved in the framework.

3.1 Graph Construction

Each event captured by an event camera records both spatial information and precise temporal details, crucial for dynamic scene analysis. That is, each event is recorded as

$$event_i = (x_i, y_i, t_i, p_i), \quad i \in \mathcal{N}, \tag{1}$$

where x_i, y_i locate the pixel and t_i represents the timestamp with microsecond precision. The coordinates x_i and y_i as well as the timestamps t_i are completely asynchronous and occur randomly, reflecting the event-driven nature of the data capture. The polarity $p_i \in \{-1, 1\}$ denotes whether there is a decrease (for -1) or an increase (for 1) in pixel brightness. $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of indexes of events.

A graph-based representation is employed to capture irregular spatiotemporal relationships between events. Firstly, in order to enhance the stability of the model and prevent data bias, especially for timestamps with otherwise large values, it is necessary to normalize the timestamp for each event by the equation $t_i^* = \beta \cdot (t_i - t_0), i \in \mathcal{N}$, where $t_0 = \min_{i \in \mathcal{N}}\{t_i\}$ and β is a normalization factor. Each event $event_i$ then generates a vertex $v_i = (x_i, y_i, t_i^*, p_i)$, and $V = \{v_1, v_2, \ldots, v_N\}$ is the set of vertices. The position coordinates of each vertex can be represented as $\mathbf{c}_i = [x_i, y_i, t_i^*]^T$.

To establish the graph edges, we consider every pair $(\mathbf{c}_i, \mathbf{c}_j)$. An edge e_{ij} is added to the set E of edges if the conditions are met,

$$e_{ij} = \begin{cases} 1, & \text{if } \|\mathbf{c}_i - \mathbf{c}_j\| \leq R, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

indicating the connectivity based on the predefined spatiotemporal distance threshold R.

The resulting graph, $G = (V, E)$, effectively encapsulates the essential spatialtemporal properties of events.

3.2 Connected Subgraphs Construction

Event cameras produce data triggered by pixel brightness changes. Dense events occur at object contours where brightness exhibits a significant variation, whereas the main body sees fewer, sparse events due to subtle changes. Transforming this event data into graph G, areas with dense events display numerous edges, whereas sparser areas show fewer edges. This pattern of data density leads us to the concept of connected graphs in graph theory, which describes the interconnectivity between the nodes of an undirected graph. The contour regions, due to their high number of edges, form several large connected subgraphs with a substantial number of nodes. In contrast, the main bodies, which have fewer edges, result in connected subgraphs with fewer nodes. By filtering the data based on the number of nodes, the majority of the retained connected subgraphs attain new physical meaning, indicating the contiguous edges of object contours. The entire graph G can be divided into K connected subgraphs, $G_k = (V_k, E_k)$, which satisfy

$$\mathcal{D} = \{G_k \colon G_k \subseteq G, \quad k = 1, 2, \cdots, K\}. \tag{3}$$

However, it should be noted that the connected subgraphs are not in one-to-one correspondence with the objects D_l in the event data. Consequently, the number of selected subgraphs, K, is typically greater than the number of real objects, L.

$$\mathcal{D}_l \subseteq \mathcal{D}, \quad l = 1, 2, \cdots, L \tag{4}$$

By filtering the number of nodes contained in the connected subgraphs, we can filter out subgraphs containing too few nodes, thus realizing downsampling. Based on preliminary experimental tests with the dataset used, processing 10,000 events can retain approximately 73% of the events, which

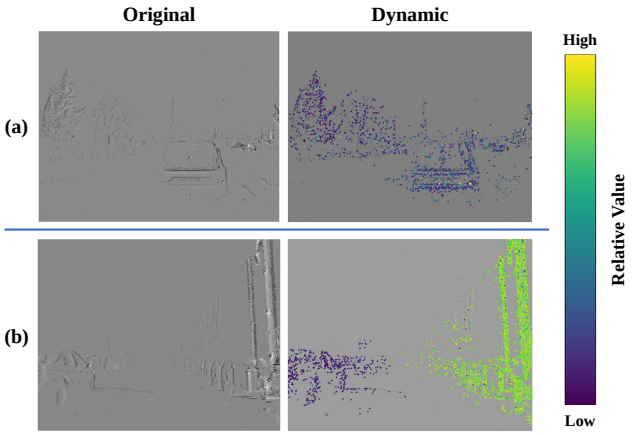

Figure 2: **Dynamic Visualization of the SSM.** Each image is generated from 10,000 event points, causing slight blurring. However, connected subgraphs effectively filter out background noise, preserving only relevant objects. (a) The scene shows low relative dynamic, hence the distinction is not pronounced. (b) The truck on the right accelerates to overtake, while the car on the left moves slower, making the truck's relative values significantly higher.

helps reduce noise interference. In contrast to conventional uniform downsampling, this approach avoids the loss of valid features associated with nodes. Moreover, the scope of event data that requires focus can be selected in a more effective and rational manner.

After constructing connected subgraphs, we utilize a multilayer GCN that integrates node feature information with graph topology, efficiently learning node representations. The GCN preserves and fuses spatial and temporal features globally, producing a global graph feature map.

3.3 Spatiotemporal Sensitivity Module (SSM)

The Spatiotemporal Sensitivity Module (SSM) is a core component of our framework. It mimics the visual characteristics of the human eye to quantify object dynamics in event data. Since event data differs from traditional images, directly quantifying object dynamics using displacement speed is challenging. Thus, we propose Event Global Motion (EGM) and Event Local Motion (ELM), applied to the global graph G and subgraphs G_k, respectively.

We first define the following metrics to separately quantify the global and local dynamics:

For global motion, we define:

$$EGM = f(N, \Delta t^*), \tag{5}$$

where $N = |\mathcal{N}|$ represents the number of nodes in the global graph, and $\Delta t^* = \max_{i,j \in \mathcal{N}}\{|t_i^* - t_j^*|\}$ represents the maximum time difference among nodes.

For local motion, we define:

$$ELM_k = f(n_k, \delta t_k^*), \tag{6}$$

where n_k represents the number of nodes involved in the measurement of the k-th subgraph, and δt_k^* is the maximum time difference within the subgraph. While the function f could be designed to take more complex forms to capture intricate relationships between these variables, we opt for a simple ratio-based method, $f(x, y) = x/y$. This choice is motivated by the desire to balance simplicity and computational efficiency. The ratio directly reflects the relationship between the number of events and the temporal spread in the subgraph, providing an intuitive measure of local motion dynamics. By using this straightforward formulation, we ensure that the computation remains tractable, especially when dealing with large-scale event data, while still effectively capturing the key dynamics of interest.

For instance, in a traffic scenario, event data may include dynamic entities such as vehicles and pedestrians, as well as static background elements like trees and fences. EGM quantifies the dynamics of the entire event data, representing the overall level of motion in the environment, while ELM_k describes the motion level of individual subgraphs. Given the high temporal resolution of

event cameras, we can assume that any object responds to events almost instantaneously. As an object's speed increases, more pixels detect brightness changes, generating more events, which increases n_k and decreases δt_k^*, leading to a significant increase in ELM_k.

To analyze the relationship between the dynamics of each subgraph relative to the overall environment, we introduce the following metric:

$$H = \left\{ h_k = \frac{ELM_k}{EGM} : k = 1, 2, \ldots, K \right\}, \tag{7}$$

which quantifies the relative motion of each subgraph with respect to the global dynamic level. During subgraph construction, distance thresholds and limitations on adjacent nodes are set to divide large objects into multiple subgraphs, thus preventing a single subgraph from dominating the motion representation of a large object.

While all nodes in each subgraph should reflect relative dynamics, assigning the same features to every node is unnecessary as it would result in redundant computations. We aggregate the spatiotemporal coordinates of each subgraph into representative coordinates as follows:

$$S = \{ s_k = \pi(V_k^*) : k = 1, 2, \ldots, K \}, \tag{8}$$

where V_k^* denotes the set of spatiotemporal position coordinates for subgraph nodes, and π is a function for aggregation [12, 41], such as mean, max, or min. These functions are chosen because they are insensitive to the varying number of nodes in each subgraph, ensuring that the aggregation process remains consistent regardless of subgraph size. We then use the K-nearest neighbors algorithm to connect these new representative coordinates, forming a new set of edges $\mathcal{E}$, mapping features h_k to the corresponding representative coordinates, and building a new set of vertices $\mathcal{V}$. This results in a new graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which significantly reduces the number of nodes and decreases computational complexity and processing delay.

In the newly constructed graph, each node represents an aggregated subgraph with a relatively sparse distribution in the spatiotemporal domain, necessitating the enhancement or suppression of relationships between nodes. We introduce inter-subgraph attention using a Graph Attention Network (GAT) [23] to capture relationships among nodes. The GAT enables nodes to adaptively aggregate information based on the importance of their features and their neighbors, strengthening the connections between nodes representing the same object while weakening those representing different objects. The GAT's attention mechanism also ensures that smaller objects are not overlooked and evaluates their significance in the overall environment. The aggregated values are used as the final output of the SSM for assessing relative motion. Figure 2 shows some example results, illustrating how the spatiotemporal sensitivity module captures the relative dynamics in different regions.

3.4 Temporal Activation Controller

Upon evaluation of the output from the SSM, a determination is made as to whether to enhance the temporal focus by activating the TAC. The TAC processing comprises two parts: firstly, a simple attention-like method is employed to dynamically weight and selectively emphasise the temporal dimension; secondly, the Query, Key, and Value (QKV) are reconstructed.

In the first part of the TAC, we define a set $\mathcal{F} = \{F_t\}$, where F_t represents the stream of graph feature maps over time t, and $\mathcal{F}$ is the collection of these streams. We aggregate all features within the set using the formula $F_{st} = Aggregation\{\mathcal{F}\}$ to generate the Global Spatiotemporal Feature, F_{st}, which serves as a comprehensive representation of the spatiotemporal data. Subsequently, we employ an Adaptive Gating Mechanism to adjust these features at each timestamp, generating gating signals from F_{st} for each F_t. This mechanism either enhances or suppresses the features based on the global feature, thereby maintaining temporal integrity while adapting to the sequence context. Finally, an adaptive fusion technique modulates the integration between the original and enhanced features, producing outputs suitable for subsequent processing.

In the second part of TAC, different sources generate the QKV matrices,

$$\begin{aligned} Q &= W_q(F_{st}), \\ K, V &= W_{k,v}(\mathcal{F}). \end{aligned} \tag{9}$$

The global spatiotemporal feature F_{st} is employed to generate the Query (Q), which offers a global perspective when assessing the relevance of each timestamp to the entire sequence. Meanwhile, the

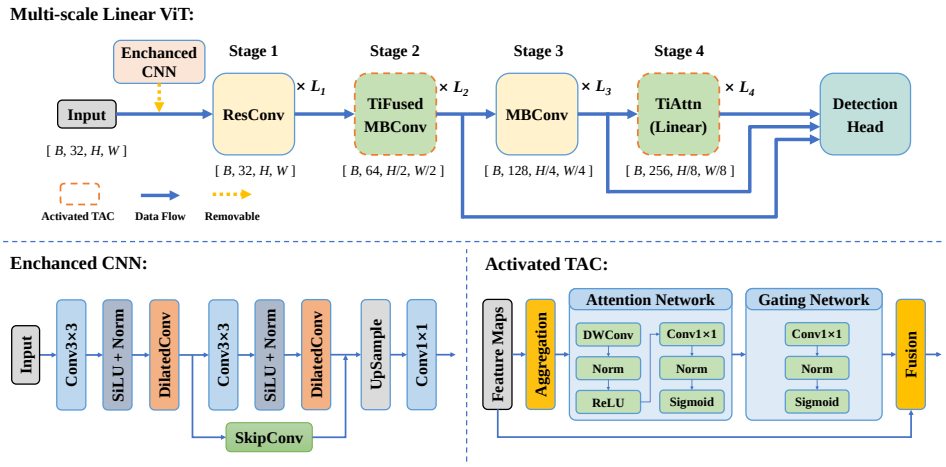

Figure 3: **The flowchart of the Multi-scale Linear ViT.** This diagram shows the stages of the Multi-scale Linear ViT, with the removable Enhanced CNN and Activated TAC modules. The Enhanced CNN processes input data through convolutional and normalization layers before passing it to the ViT stages. Activated TACs at Stage 2 and Stage 4 optimize temporal processing and balance efficiency. The data is then sent to the detection head for final object detection.

utilisation of the aggregated outputs of each timestamp in the first part as the source of Keys (K) and Values (V) generation helps to reflect the state of each timestamp with greater accuracy. This method of employing diverse generation sources merges global and local insights, thereby bolstering the model's capability to process complex spatiotemporal data within a self-attention architecture.

It is worth mentioning that TAC is not a module but a component, so it can theoretically be loaded on any model focusing on the extraction of spatial features, greatly improving scalability.

3.5 Multi-scale Linear ViT

The original Vision Transformer (ViT) [42] does not allow for multi-scale processing, resulting in significant performance degradation compared to models equipped with such capabilities [43, 44, 45]. To address this issue, our enhanced ViT module supports multi-scale processing and incorporates the TAC, which improves the model's ability to handle spatial and temporal dimensions. To reduce the computational load and increase processing speed, we adopt linear transformations to simplify the self-attention mechanism. Drawing upon techniques from models like Efficient ViT [24], which reduce the computational complexity of self-attention from $O(N^2)$ to approximately $O(N)$, our ViT module adapts these cost-reduction strategies to fit within our framework.

4 Experiments

In this section, we introduce the two datasets utilized, the evaluation metrics, and the implementation details of our models. We train the baseline model, EGSST-B, and the extended model, EGSST-E, and compare their performance with other state-of-the-art models applied to both datasets. Detailed ablation studies are then performed to assess the impact of various components of our models. Finally, to verify the scalability of the models, we train them using varying numbers of events, obtain their corresponding weights, and analyze these weights in differently configured models.

4.1 Datasets and Evaluation Metrics

Two complex event camera datasets from traffic scenarios are employed in the experiments: the Gen1 Automotive Detection Dataset [46] and the 1 Megapixel Automotive Detection Dataset [9].

The Gen1 Dataset comprises over 39 hours of event video from urban, highway, and rural settings, with a resolution of 304×240 pixels. The dataset includes manual annotations of pedestrians and

cars, with over 255,000 labels at a frequency of 1 to 4 Hz, making it ideal for testing object detection, tracking, and optical flow algorithms in automotive environments.

The 1Mpx Dataset comprises over 14 hours of high-resolution (1 megapixel) event video, annotated with 25 million labels for cars, pedestrians, and two-wheelers, suitable for developing advanced detection models in dynamic traffic conditions.

Three primary evaluation metrics are employed in the experiments, namely the total number of parameters, the mean Average Precision (mAP@0.5:0.95) using the COCO toolbox [47], and the mean time per batch (batch size = 1) for detection. These metrics assess the models' complexity, precision, and efficiency, respectively, in real-time applications.

4.2 Implementation Details

The framework proposed in this study is developed using Python 3.9 and PyTorch 2.0, with graph processing powered by the advanced PyTorch Geometric library [48]. To enhance the scalability and parallelism of subgraph processing, modifications are made to the underlying libraries to achieve complete parallelization of graph processing, thus fully leveraging GPU computational capabilities. The models are trained on RTX3090 GPUs using the Lightning framework. We employ the Adam optimizer [49] coupled with the OneCycle learning rate schedule [50], which includes 100 warm-up iterations followed by cosine decay starting from the maximum learning rate. The training batch size is set at 8, with an initial learning rate of 1e-6.

4.3 Dynamic Label Augmentation

In our model, we use a batch segmentation method based on a fixed number of events, which requires generating corresponding labels for inputs with varying numbers of nodes. However, the data generation speed of event cameras is extremely high, while the number of labels available in existing datasets is relatively limited. For instance, current datasets such as Gen1 cannot effectively provide sufficient labels to match the rapidly collected fixed-number event data. To address this mismatch, it is necessary to develop an effective approach to increase the number of available labels.

A common approach is to extend the labeling by adding a fixed time window before and after each label, mapping all the data within that time frame to the same label [12, 27, 51, 52]. While this method can indeed increase the number of labels under typical conditions, it can introduce significant labeling errors when applied to fixed-number event-based batch segmentation, especially in dynamic environments with substantial variability.

To address this issue, we propose a dynamic label augmentation method. This approach dynamically adjusts the labeling precision based on the time span over which events are captured, aiming to expanding the number of labels while enhancing labeling accuracy. Specifically, in the dynamic label augmentation method, a longer time span for collecting a fixed number of events indicates slower target motion, allowing for an expanded time window to capture more labels for the current target. Conversely, a shorter time span suggests faster target motion, necessitating a smaller time window to maintain accurate labeling. More details can be found in the Appendix C.

4.4 Benchmark Comparisons

Our baseline model, EGSST-B, achieved an impressive processing time of only 2.4 milliseconds on the RTX 3090. However, this result was obtained without considering GPU power consumption. To ensure a fair comparison with other models, we conducte additional tests on the T4 GPU, which has performance comparable to the Titan Xp and RTX 1080Ti. The ASTMNet and RED models are tested on the Titan Xp, while the AEC model was evaluated on the RTX 1080Ti.

Table 1 compares the performance of various target detection methods on the Gen1 and 1 Mpx datasets. Among these benchmarks, our models EGSST-B and EGSST-E outperform the rest. EGSST-E achieves a 49.6% mAP on the 1 Mpx dataset, demonstrating exceptional capabilities; EGSST-B processes in just 4.6 milliseconds on the Gen1 dataset, significantly outperforming other models in terms of efficiency and real-time processing. With parameter counts of 3.5M and 12.3M, respectively, these models achieve high performance while being more streamlined compared to traditional approaches. This optimization in parameter efficiency is particularly crucial for deployments on resource-limited platforms.

Table 1: **Comparison of Different Event-Based Vision Methods.** The results of our methods are obtained from experiments involving 10,000 events. Our methods ending in -Y utilize the YOLOX [26] detection head instead of the RT-DETR [25] method. A star * indicates that this information is not directly available and can be estimated based on modules in published articles.

Method	Backbone	Gen1		1 Mpx		Params (M)
		mAP (%) ↑	Time (ms) ↓	mAP (%) ↑	Time (ms) ↓	
RRC-Events [32]	CNN	30.7	21.5	34.3	46.4	>100*
AEGNN [12]	GNN	16.3	-	-	-	20.0
Spiking DenseNet [20]	SNN	18.9	-	-	-	8.2
ERGO-12 [53]	Transformer	**50.4**	69.9	40.6	100.0	59.6
RED [9]	CNN + RNN	40.0	16.7	43.0	39.3	24.1
ASTMet [37]	CNN + RNN	46.7	35.6	48.3	72.3	>100*
AEC + DETR [54]	-	44.5	7.7	45.9	20.7	>40*
AEC + YOLOv5 [54]	-	47.0	<u>3.9</u>	<u>48.4</u>	13.8	>40*
RVT-B [27]	Transformer + RNN	47.2	10.2	47.4	11.9	18.5
GET-T [52]	Transformer + RNN	47.9	16.8	<u>48.4</u>	18.2	21.9
S4D-ViT-B [51]	Transformer + SSM	46.2	9.4	46.8	10.9	16.5
S5-ViT-B [51]	Transformer + SSM	47.4	8.16	47.2	9.57	18.2
EGSST-B (ours)	GNN + LinearViT	44.6	4.6	45.4	<u>5.1</u>	3.5
EGSST-E (ours)	GNN + LinearViT	<u>49.6</u>	6.0	**50.2**	6.3	12.3
EGSST-B-Y (ours)	GNN + LinearViT	43.9	**3.7**	44.1	**5.0**	3.1
EGSST-E-Y (ours)	GNN + LinearViT	47.8	4.2	48.3	5.3	10.4

The EGSST model leverages a fully event-based architecture that makes full use of the rich spatiotemporal data generated by event cameras. The model achieves low processing latency through effective data downsampling, efficient batch processing, and the use of techniques that mimic the dynamic characteristics of the human eye. Additionally, the application of linear feature extractors further enhances its processing efficiency. These aspects enable the model to effectively utilize global information while minimizing computational demands, making it well-suited for object detection tasks using event cameras.

4.5 Ablation Studies

All experiments are conducted under identical environmental and equipment conditions. Given the similar performance observed on the 1Mpx and Gen1 datasets, the experiments described in this section are based on the Gen1 dataset unless otherwise stated. Moreover, since both of our models demonstrated similar test results, only the EGSST-E model is used for further analysis.

Table 2: **Impact of applying TAC.** The 'TAC Adaptive' refers to the integration of SSM with TAC, allowing for adaptive adjustments based on the data state.

Condition	mAP (%)	Time (ms)	Params (M)
TAC Inactive	42.1	4.3	10.2
TAC Active	51.5	11.1	16.3
TAC Adaptive	49.6	6.0	12.3

The SSM and TAC, integrated in series within our framework, are the reason why we conduct comparative experiments with these two components together. The presence of SSM dynamically activates TAC, thereby causing the parameters involved in the forward computation to vary dynamically as well. For quantitative analysis, we calculated the average amount of parameters under the condition of 10,000 event inputs across our test dataset. As shown in Table 2, the application state of TAC significantly impacts all evaluation metrics. By activating TAC at appropriate times, SSM avoids unnecessary computations and time loss, which not only enhances model efficiency but also confirms the effectiveness of SSM. Although the full application of TAC can maximize mAP accuracy, the additional time and computational costs involved are not economical.

Table 3: **Comparison after incorporating CNN.** The increase with the addition of the CNN is shown in parentheses.

Method (with CNN)	mAP (%)	Time (ms)	Params (M)
EGSST-B	44.6 (+0.4)	4.6 (+0.1)	3.51 (+0.02)
EGSST-E	49.6 (+0.5)	6.0 (+0.1)	12.34 (+0.02)

Removable Convolutional Neural Network. Our framework is event-based and does not use CNN for feature extraction. Nevertheless, it has been observed that integrating a simple CNN module enhances detection performance. To operationalize this module, a certain number of event data must be standardized across the temporal dimension to form an image-like representation. Table 3 shows that this CNN integration increases mAP with minimal impact on processing time and parameters. Consequently, we have incorporated a removable CNN layer into the model, allowing for flexibility to revert to a fully event-driven configuration if needed.

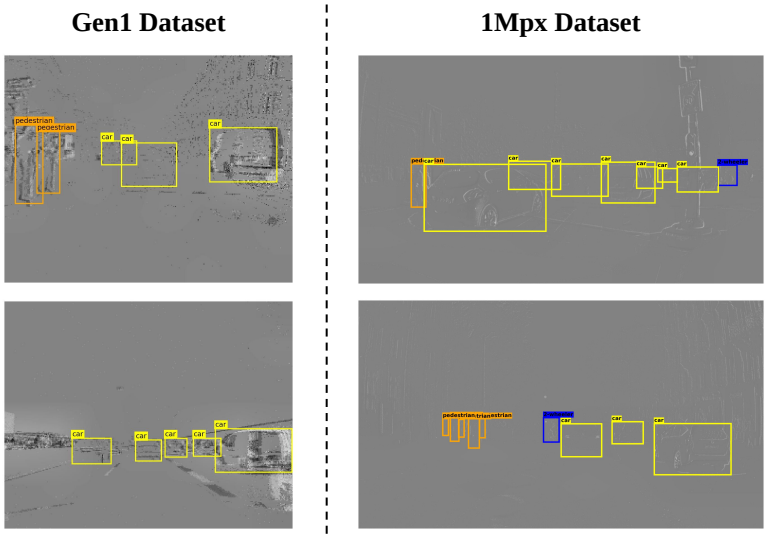

Figure 4: **Prediction Results.** Due to the low accumulated event count, the visualizations appear somewhat blurred. Nevertheless, our model effectively identifies objects within these sparse events, demonstrating its robustness and efficacy.

5 Discussion and Conclusion

We present a lightweight, efficient, and accurate event-based object detection framework tailored for event cameras. Our model shows strong detection performance and scalability, highlighting its practical application potential. Future work will extend this approach to object tracking and other advanced visual tasks, as well as explore deployment optimizations, such as enhancing execution speeds and converting models to ONNX or TensorRT formats. Deploying ViTs is relatively straightforward, while GNNs pose greater challenges due to complex graph processing requirements. Although progress has been made, further work is necessary to improve the deployment efficiency of GNNs.

Furthermore, we plan to investigate the integration of event data with RGB frames in a multimodal model [55, 56, 57], which could enhance the robustness and accuracy of visual recognition tasks under varied and dynamic conditions. This multimodal approach promises to leverage the strengths of both sensor types to deliver superior performance in complex environments.

Acknowledgments and Disclosure of Funding

This research was supported by the National Key Research and Development Program under grant number 2024YFE0200703.

References

[1] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6(3):4947–4954, 2021.

[2] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1731–1740, 2018.

[3] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039, 2018.

[4] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022. doi: 10.1109/TPAMI.2020.3008413.

[5] Bongki Son, Yunjae Suh, Sungho Kim, Heejae Jung, Jun-Seok Kim, Changwoo Shin, Keunju Park, Kyoobin Lee, Jinman Park, Jooyeon Woo, et al. 4.1 a 640× 480 dynamic vision sensor with a 9μm pixel and 300meps address-event representation. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 66–67. IEEE, 2017.

[6] Daniel Gehrig, Antonio Loquercio, Konstantinos G Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5633–5643, 2019.

[7] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7243–7252, 2017.

[8] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza. Learning monocular dense depth from events. In *2020 International Conference on 3D Vision (3DV)*, pages 534–542. IEEE, 2020.

[9] Etienne Perot, Pierre De Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. Learning to detect objects with a 1 megapixel event camera. *Advances in Neural Information Processing Systems*, 33:16639–16652, 2020.

[10] Stepan Tulyakov, Daniel Gehrig, Stamatios Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza. Time lens: Event-based video frame interpolation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16155–16164, 2021.

[11] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[12] Simon Schaefer, Daniel Gehrig, and Davide Scaramuzza. Aegnn: Asynchronous event-based graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12371–12381, 2022.

[13] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021.

[14] Thomas Dalgaty, Thomas Mesquida, Damien Joubert, Amos Sironi, Pascal Vivet, and Christoph Posch. Hugnet: Hemi-spherical update graph neural network applied to low-latency event-based optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3952–3961, 2023.

[15] Dominik Eisl, Fabian Herzog, Jean-Luc Dugelay, Ludovic Apvrille, and Gerhard Rigoll. Introducing a framework for single-human tracking using event-based cameras. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 3269–3273. IEEE, 2023.

[16] Samanwoy Ghosh-Dastidar and Hojjat Adeli. Spiking neural networks. *International journal of neural systems*, 19(04):295–308, 2009.

[17] Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training deep spiking neural networks using backpropagation. *Frontiers in neuroscience*, 10:228000, 2016.

[18] Sumit B Shrestha and Garrick Orchard. Slayer: Spike layer error reassignment in time. *Advances in neural information processing systems*, 31, 2018.

[19] Mathias Gehrig, Sumit Bam Shrestha, Daniel Mouritzen, and Davide Scaramuzza. Event-based angular velocity regression with spiking networks. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4195–4202. IEEE, 2020.

[20] Loïc Cordone, Benoît Miramond, and Philippe Thierion. Object detection with spiking neural networks on automotive event data. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.

[21] Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, and Xin Yang. Spiking transformers for event-based single object tracking. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 8801–8810, 2022.

[22] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[23] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[24] Han Cai, Junyan Li, Muyan Hu, Chuang Gan, and Song Han. Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17302–17313, October 2023.

[25] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16965–16974, 2024.

[26] Z Ge. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[27] Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13884–13893, 2023.

[28] Xu Zheng, Yexin Liu, Yunfan Lu, Tongyan Hua, Tianbo Pan, Weiming Zhang, Dacheng Tao, and Lin Wang. Deep learning for event-based vision: A comprehensive survey and benchmarks. *arXiv preprint arXiv:2302.08890*, 2023.

[29] Cedric Scheerlinck, Henri Rebecq, Timo Stoffregen, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Ced: Color event camera dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[30] Patrick Bardow, Andrew J Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 884–892, 2016.

[31] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. A differentiable recurrent surface for asynchronous event-based data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 136–152. Springer, 2020.

[32] Nicholas FY Chen. Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 644–653, 2018.

[33] Massimiliano Iacono, Stefan Weber, Arren Glover, and Chiara Bartolozzi. Towards event-driven object detection with off-the-shelf deep learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE, 2018.

[34] Zhuangyi Jiang, Pengfei Xia, Kai Huang, Walter Stechele, Guang Chen, Zhenshan Bing, and Alois Knoll. Mixed frame-/event-driven fast pedestrian detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8332–8338. IEEE, 2019.

[35] Guillermo Gallego, Jon EA Lund, Elias Mueggler, Henri Rebecq, Tobi Delbruck, and Davide Scaramuzza. Event-based, 6-dof camera tracking from photometric depth maps. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2402–2412, 2017.

[36] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 934–943, 2021.

[37] 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*, 31:2975–2987, 2022.

[38] Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-based asynchronous sparse convolutional networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 415–431. Springer, 2020.

[39] Ze Huang, Li Sun, Cheng Zhao, Song Li, and Songzhi Su. Eventpoint: Self-supervised interest point detection and description for event-based camera. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5396–5405, 2023.

[40] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. Asynchronous convolutional networks for object detection in neuromorphic cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[41] Yanxiang Wang, Xian Zhang, Yiran Shen, Bowen Du, Guangrong Zhao, Lizhen Cui, and Hongkai Wen. Event-stream representation for human gaits identification using deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3436–3449, 2021.

[42] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[43] Wenhai Wang, Enze Xie, Xiang Li, Dengping Fan, Cheng Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021.

[44] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[45] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021.

[46] 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.

[47] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[48] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[49] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[50] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019.

[51] Nikola Zubic, Mathias Gehrig, and Davide Scaramuzza. State space models for event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5819–5828, June 2024.

[52] Yansong Peng, Yueyi Zhang, Zhiwei Xiong, Xiaoyan Sun, and Feng Wu. Get: Group event transformer for event-based vision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6038–6048, October 2023.

[53] Nikola Zubić, Daniel Gehrig, Mathias Gehrig, and Davide Scaramuzza. From chaos comes order: Ordering event representations for object recognition and detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12846–12856, October 2023.

[54] Yansong Peng, Yueyi Zhang, Peilin Xiao, Xiaoyan Sun, and Feng Wu. Better and faster: Adaptive event conversion for event-based object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2056–2064, 2023.

[55] Jianing Li, Siwei Dong, Zhaofei Yu, Yonghong Tian, and Tiejun Huang. Event-based vision enhanced: A joint detection framework in autonomous driving. In *2019 ieee international conference on multimedia and expo (icme)*, pages 1396–1401. IEEE, 2019.

[56] Dianze Li, Yonghong Tian, and Jianing Li. Sodformer: Streaming object detection with transformer using events and frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):14020–14037, 2023.

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

A Efficient ViT

This section extends section 3.5 by explaining the methodologies used in our architecture with Linear ViT, which originates from [24] and is a vital component of our model. For further details, readers are referred to the original paper.

Traditional ViTs are not suitable for fast visual tasks due to their reliance on extensive softmax attention. Softmax attention modules model interactions between every pair of tokens in the feature map to aggregate spatial information, leading to high computational complexity. Efficient ViT is a speedy visual model that replaces the computationally intense softmax calculations with multi-scale linear attention, maintaining excellent hardware efficiency while achieving a global receptive field and multi-scale learning.

Firstly, to reduce computational complexity and hardware latency, softmax attention is replaced with linear attention, formulated as:

$$O_i = \sum_{j=1}^{N} \frac{\text{Sim}(Q_i, K_j)}{\sum_{j'=1}^{N} \text{Sim}(Q_i, K_{j'})} V_j \tag{10}$$

where

$$\text{Sim}(Q, K) = \text{ReLU}(Q)\text{ReLU}(K)^T. \tag{11}$$

This leads to the final output:

$$O_i = \frac{\text{ReLU}(Q_i)\left(\sum_{j=1}^{N}\text{ReLU}(K_j)^T V_j\right)}{\text{ReLU}(Q_i)\left(\sum_{j=1}^{N}\text{ReLU}(K_j)^T\right)}, \tag{12}$$

significantly reducing both computational complexity and memory requirements.

However, due to the loss of local feature extraction capability by ReLU, linear attention's performance lags behind softmax attention. To improve performance, two core solutions have been proposed:

1. Insert deep convolutions in each FFN layer. Linear attention extracts global features, while FFN+DWConv captures local information, minimizing computational overhead and greatly enhancing local feature extraction capability of linear attention.
2. By aggregating adjacent Q, K, V token information into multi-scale tokens, the multi-scale learning capability across different channels of linear attention is enhanced.

B Performance Scalability Analysis

To test the scalability of our models, we use inputs with three different event counts to train distinct weight parameters and conducted cross-testing under various input conditions. Given the similar performance outcomes of EGSST-B and EGSST-E, we present specific results only for EGSST-E.

Table 4: **Performance scalability analysis with different number of input events.** The results here are all run on the Gen1 dataset and the results on 1Mpx are similar. (Note: T refers to thousand.)

	EGSST-E (2T)		EGSST-E (10T)		EGSST-E (18T)		EGSST-E
Events	mAP (%) ↑	Time (ms) ↓	mAP (%) ↑	Time (ms) ↓	mAP (%) ↑	Time (ms) ↓	Params (M)
2,000	34.9	4.6	39.4	4.6	37.7	4.6	12.3
10,000	33.1	6.1	49.6	6.0	44.5	6.2	12.3
18,000	30.6	7.9	45.4	7.9	51.7	7.8	12.3

In terms of runtime, it can be demonstrated that a higher number of events will invariably lead to a higher consumption of time. In terms of mAP, cross-test results demonstrate that model weights trained with fewer events perform well when applied to datasets with a larger number of events, highlighting the excellent generalization capabilities of our models. Conversely, although models trained with a larger number of events show some performance disparity when applied to datasets with fewer events, the performance remains acceptable, indicating good adaptability of the model.

Our models allow for the flexible setting of the event count N, leveraging certain capabilities from dynamic graph processing in the treatment of GCN. While a larger event count N introduces more node features, significantly enhancing detection accuracy, it also results in increased computational load and latency.

C Dynamic Label Augmentation

C.1 Methodology

In this paper, we propose a novel dynamic label matching method to address the limitations of traditional label matching approaches when processing fixed numbers of event data. Specifically, the event data acquisition speed is extremely high, while existing datasets typically lack a sufficient number of labels to match the accumulated fixed-number event data. Therefore, traditional methods based on fixed time window label assignment are inadequate for scenarios requiring both adaptability to dynamic scenes and sufficient label coverage. To overcome this limitation, we design a Dynamic Label Augmentation approach to flexibly assign appropriate labels for each batch of fixed-number events.

In the proposed method, we assume that the fixed number of collected events is 10,000, for which labels need to be assigned. Suppose there are m labels in the original dataset, with each label corresponding to a timestamp τ_m. For each batch of 10,000 events, we first compute the representative time, defined as the mean timestamp of these events, denoted by $\bar{t}_r$, where r represents the index of the current batch of 10,000 events.

To achieve dynamic label matching, we introduce a constant γ to control the flexibility of the label matching time window. Specifically, we define a dynamically adjusted time range $\gamma \cdot \Delta t^*$, where Δt^* is a time parameter that varies in real-time based on the dynamics of the events. A label with a timestamp τ_m is considered appropriate for the current batch of 10,000 events if it satisfies the following condition:

$$\bar{t}_r - \gamma \cdot \Delta t^* \leq \tau_m \leq \bar{t}_r + \gamma \cdot \Delta t^* \tag{13}$$

Since Δt^* changes dynamically according to the characteristics of the events, the proposed method can effectively adapt to variations in the event data, ensuring the accuracy of label assignment.

In summary, the Dynamic Label Augmentation method enables flexible adjustment of the time window, allowing the label assignment process to adapt to different dynamic properties of the events. This approach avoids potential mismatches associated with traditional fixed time window methods and exhibits strong robustness and adaptability.

C.2 Experiment

Table 5 illustrates the impact of various data augmentation techniques on model accuracy, with dynamic label augmentation demonstrating a relative advantage. Traditional augmentation methods, such as horizontal flipping, zooming in, and zooming out, improve model robustness by increasing data diversity, contributing to accuracy improvements. However, dynamic label augmentation, which adaptively adjusts the label generation range, shows better adaptability when handling dynamic scenes, particularly by reducing the risk of label mismatches, thereby potentially enhancing overall model performance.

Table 5: **Accuracy Improvement from Dynamic Label Augmentation.** All augmentation techniques improve accuracy, with dynamic augmentation showing the greatest improvement.

h-flip	zoom-in	zoom-out	dynamic	mAP (%)
				38.3
✓				41.8
	✓			43.3
		✓		42.1
			✓	46.6
✓	✓	✓	✓	49.6

The experimental results suggest that while conventional augmentation techniques are effective in improving accuracy, dynamic label augmentation may provide additional gains in both accuracy and robustness by flexibly adjusting the label generation process. This approach appears to improve the model's ability to adapt to complex and fast-changing environments to a certain extent.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately reflect the paper's contributions and scope because they succinctly summarize the key methodology employed in the research, providing a clear overview that aligns with the detailed content presented in the subsequent sections of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations of the work performed by the authors in section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This is an applied paper that provides an event-based processing framework, and as such, it does not focus on theoretical results that require detailed assumptions and proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper fully discloses all the necessary information to reproduce the main experimental results by providing detailed descriptions of the methodologies and parameters used, ensuring that the main claims and conclusions can be independently reproduced.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper provides open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in the supplemental material, and the code is included in the attachment.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper specifies all the training and test details necessary to understand the results in section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experimental results in the paper do not include error bars or other information about the statistical significance of the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides sufficient information on the computer resources needed to reproduce the experiments in section 4.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms to the NeurIPS Code of Ethics as it ensures the responsible use of data, respects participant privacy and commits to integrity and ethical standards in research.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: There is no societal impact of the work performed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the original papers or websites that produced the code package or dataset.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This is an applied paper on event camera applications, and it does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

Justification: The paper does not involve human experiments or study participants, so there are no potential risks to describe, no disclosures to subjects, and no need for IRB approvals.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.