Neuromorphic Camera Denoising Using Graph Neural Network-Driven Transformers

Yusra Alkendi, Rana Azzam, Abdulla Ayyad, *Member, IEEE*, Sajid Javed, Lakmal Seneviratne, and Yahya Zweiri, *Member, IEEE*

***Abstract*— Neuromorphic vision is a bio-inspired technology that has triggered a paradigm shift in the computer vision community and is serving as a key enabler for a wide range of applications. This technology has offered significant advantages, including reduced power consumption, reduced processing needs, and communication speedups. However, neuromorphic cameras suffer from significant amounts of measurement noise. This noise deteriorates the performance of neuromorphic event-based perception and navigation algorithms. In this article, we propose a novel noise filtration algorithm to eliminate events that do not represent real log-intensity variations in the observed scene. We employ a graph neural network (GNN)-driven transformer algorithm, called GNN-Transformer, to classify every active event pixel in the raw stream into real log-intensity variation or noise. Within the GNN, a message-passing framework, referred to as EventConv, is carried out to reflect the spatiotemporal correlation among the events while preserving their asynchronous nature. We also introduce the known-object ground-truth labeling (KoGTL) approach for generating approximate ground-truth labels of event streams under various illumination conditions. KoGTL is used to generate labeled datasets, from experiments recorded in challenging lighting conditions, including moon light. These datasets are used to train and extensively test our proposed algorithm. When tested on unseen datasets, the proposed algorithm outperforms state-of-the-art methods by at least 8.8% in terms of filtration accuracy. Additional tests are also conducted on publicly available datasets (ETH Zürich Color-DAVIS346 datasets) to demonstrate the generalization capabilities of the proposed algorithm in the presence of illumination variations and different motion dynamics. Compared to state-of-the-art solutions, qualitative results verified the superior capability of the proposed algorithm to eliminate noise while preserving meaningful events in the scene.**

Manuscript received 23 November 2021; revised 18 April 2022 and 4 July 2022; accepted 22 August 2022. This work was supported in part by the Khalifa University of Science and Technology under Award RC1-2018-KUCARS and in part by the Aerospace Research and Innovation Center (ARIC), which is jointly funded by STRATA Manufacturing PJSC (a Mubadala company) and the Khalifa University of Science and Technology. The work of Sajid Javed was supported by the Khalifa University of Science and Technology under the Faculty Start Up Grant FSU-2022-003 Award No. 8474000401. *(Corresponding author: Yusra Alkendi.)*

Yusra Alkendi and Yahya Zweiri are with the Khalifa University Center for Autonomous Robotic Systems (KUCARS) and the Department of Aerospace Engineering, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates (e-mail: yusra.alkendi@ku.ac.ae; yahya.zweiri@ku.ac.ae).

Rana Azzam, Abdulla Ayyad, and Lakmal Seneviratne are with the Khalifa University Center for Autonomous Robotic Systems (KUCARS), Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates.

Sajid Javed is with the Khalifa University Center for Autonomous Robotic Systems (KUCARS) and the Department of Electrical Engineering and Computer Science, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3201830.

Digital Object Identifier 10.1109/TNNLS.2022.3201830

***Index Terms*— Background activity (BA) noise, dynamic vision sensor (DVS), event camera, event denoising (ED), graph neural network (GNN), spatiotemporal filter, transformer.**

I. Introduction

OVER the last decade, advances in image sensor technologies have rapidly progressed, providing several alternative solutions for scene perception and navigation. The neuromorphic event-based camera also known as dynamic vision sensor (DVS) is an asynchronous sensor that mimics the neurobiological architecture of the human retina. It has caused a paradigm shift in vision algorithms due to the way visual data are acquired and processed. Instead of capturing image frames as conventional cameras, event-based cameras report asynchronous temporal differences in the scene and form a continuous stream of events, which is generated when the log intensity of each pixel changes (i.e., events) in the order of microseconds (μs). The event-based camera has the capability to overcome the limitations of conventional cameras by providing data at low latency (20 μs), high temporal resolution (>800 kHz), high dynamic range (HDR, 120 dB), and no motion blur [1]. These sensors are able to operate in a wide range of challenging illumination environments (i.e., low light conditions) while consuming an extremely low amount of power, e.g., 10–30 mW [1].

Recently, event-based cameras have been successfully employed to perform challenging tasks, such as object tracking [2], object recognition [3], monitoring [4], depth estimation [5], optical flow estimation [6], HDR image reconstruction [7], segmentation [8], guidance [9], [10], and simultaneous localization and mapping (SLAM) [11]. In the literature, the performance of such event-based applications degrades in the presence of noise [1]. The noise associated with the generated event data using DVS could be due to the lighting conditions, motion dynamics in the scene, or sensor parameters. Extraction of meaningful event data in the presence of noise is considered a major challenge and needs further developments as mentioned in [1].

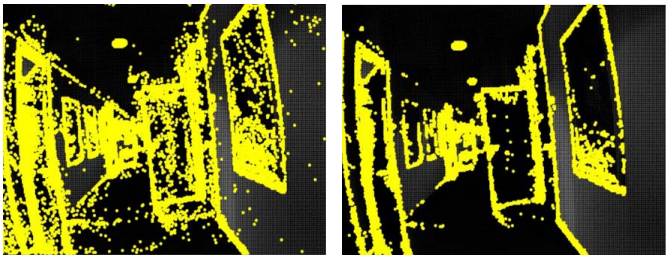
In poor lighting conditions, events corresponding to features or edges of moving objects are highly scattered and an overwhelming amount of noise is present even if optimal camera parameters are used [9], [11]. Due to the humongous amounts of events generated by DVS, manually identifying and filtering noise out is a challenging task, and therefore, research efforts are needed, especially toward noise identification and filtration in the presence of challenging lighting variations. To date, a mathematical model that accurately describes the noise associated with event streams is not yet formulated. To circumvent such a challenge, machine learning approaches can be employed to approximately model and characterize the noise parameters and consequently filter out events that

do not correspond to real intensity variations in the scene. However, the lack of labeled datasets to train event-denoising (ED) models has hindered the progress of machine learning solutions to this problem. In this article, we propose known-object ground-truth labeling (KoGTL) approach for generating approximate ground-truth labels for event streams. This is directed toward developing an ML-based ED technique that inherently copes with the nonlinear behavior of the noise associated with events.

Graph neural networks (GNNs) have shown excellent progress in a plethora of applications [13], [14]. GNN operates on data structures in the non-Euclidean domain, and hence, it is considered part of the geometric deep learning framework. In particular, GNNs operate on graphs that model a group of objects referred to as nodes and their relationships, which are referred to as edges [15]. Such data structures are not supported by conventional deep neural networks (DNNs), convolutional neural networks (CNNs), or recurrent neural networks (RNNs). GNN preserves the structure of the input graph and exploits the knowledge of the dependencies between the nodes to infer knowledge about the data. Hence, we exploit this feature of GNN and propose to design a message-passing GNN model that can operate on event streams, preserve the asynchronous nature of events, and learn to solely outflow the noise-free DVS events.

Recently, the transformers have attained significant attention in the machine learning community [16]. Vaswani *et al.* [17] proposed to model the sequence-to-sequence learning task using transformer. The self-attention mechanism within the transformer captures the relationships between input and output data and supports parallel processing of sequence recurrent networks. Transformers have recently been employed in many applications, including natural language processing and computer vision, to name a few [16], [18], [19]. In this work, we employ transformers within the proposed GNN for the task of identifying and eliminating the noise associated with events generated by DVS. To the best of our knowledge, no such research study exists in the literature where GNNs are employed together with transformers for event-based applications.

We propose a novel ED model that can learn spatiotemporal correlations between newly arrived events and the previous active events in the same neighborhood. This is achieved by means of a GNN-Transformer that operates on event streams encoded into graph structures. Our proposed algorithm consists of a message-passing GNN model and a transformer network to perform binary classification of events into real-activity events or noise. The proposed GNN-Transformer-based ED algorithm has the following advantages: 1) it can seamlessly operate on raw event streams without any data preprocessing or camera parameters' tuning; 2) it can efficiently perform in illumination conditions ranging from good light conditions to near darkness conditions; and 3) it shows robustness against different motion dynamics. The proposed GNN-Transformer is an accurate and general learning-based spatiotemporal event filter that outperforms existing denoising methods [20], [21], [22], [23], [24] in various testing scenarios. Through several tests on publicly available datasets [12], the proposed model has proven its effectiveness and capability to denoise incoming streams of events under challenging conditions in terms of illuminations and motion dynamics. Fig. 1 shows sample denoising results obtained when our proposed algorithm was used on a publicly available dataset recorded in low light conditions [12]. Our proposed algorithm operates on event graphs constructed from the incoming raw event streams where nodes represent the event properties (pixel location and time of arrival). The node of interest, i.e., the event that has just been observed, is connected through edges to the rest of the nodes that represent recent activity in the neighborhood. Then, node features are processed to generate seven messages that are sent out along the graph edges in preparation for inference and event classification. Messages are then aggregated to form a graph signature, based on which the node of interest is classified into real-activity event or noise. Since classification is done based on the graph signature rather than the raw node features, the proposed algorithm has achieved generalization across various testing datasets.



(a) (b)

Fig. 1. Denoising results using IndoorsCorridor publicly available dataset in low light scenario [12]. Events (yellow dots) are overlaid on the corresponding APS image for visualization. (a) Raw DVS stream of events and (b) Denoised events using the proposed learning-based method (GNN-Transformer). Our GNN-Transformer performs a binary classification to distinguish between actual DVS events and noise. Note that our proposed algorithm does not use APS images for denoising. All events that do not correspond to edges but are visible in the APS image have been filtered out. Our GNN-Transformer performs significantly better than the state-of-the-art methods in challenging lighting conditions (i.e., low light).

To train and test the proposed model, we develop an experimental protocol to acquire event streams from motion in different directions and under various lighting conditions. The proposed KoGTL approach is used to label events as real-activity events (class 1) or noise (class 0). The training dataset is then constructed using graph samples that encode event features and neighborhood properties, and their corresponding labels generated using KoGTL. It is worth noting that the proposed algorithm accepts input graphs of variable sizes, i.e., varying number of events in a particular spatiotemporal neighborhood. This property of the proposed ED method is very crucial since it allows for coping with the asynchronous nature of event acquisition. Experimental evaluations on various training and testing datasets demonstrate the excellent performance of the proposed algorithm compared to the existing state-of-the-art methods. The main contributions of this work are given as follows.

1) We introduce a novel KoGTL approach to generate a labeled dataset of noise and real-activity events. This dataset includes varied lighting conditions and relative motions in the visual scene.
2) We design a novel message-passing framework, dubbed EventConv, on graphs constructed from DVS events. Messages encapsulate the spatiotemporal properties of events in a neighborhood while accounting for the asynchronous nature of data acquisition.
3) We develop a novel ED GNN-Transformer architecture based on the novel EventConv layer to distinguish between real-activity and noise events.

4) We perform extensive evaluations of the proposed algorithm on our labeled dataset and other publicly available event datasets. Experiments are conducted to validate the proposed model's generalization capabilities on unseen data involving different motion dynamics and challenging lighting conditions.
5) We release a new dataset (ED-KoGTL) with labeled neuromorphic camera events acquired from motions in different directions and under various illumination conditions. Our labeled dataset is publicly available to the research community <https://github.com/Yusra-alkendi/ED-KoGTL> for benchmark comparison.

The rest of this article is organized as follows. In Section II, we review related work. In Section III, we describe the proposed algorithm and dataset in detail. The experimental results are presented in Section IV. Finally, the conclusions are drawn in Section V.

II. Background and Related Works

A. Event Denoising

The importance of the ED module to event-based computer vision algorithms has been demonstrated through several research works, such as for object recognition [25], object tracking [26], image reconstruction [26], and segmentation [27]. DVS produces noise due to various reasons. Noise could be generated due to thermal noise and junction leakage currents under constant lighting conditions. This type of noise is referred to as background activity (BA) noise. False negative events also generate noise and occur when there is no change in the log intensity. Furthermore, when a sudden change in illumination happens, a huge amount of random noise occurs in the event stream.

The BA events differ from real-activity events. BA lacks temporal correlation with the newly arrived events in the spatial neighborhood while real-activity events show meaningful correlation. Several event noise reduction methods have been proposed in the literature. These methods can be categorized into conventional methods [21], [22], [23], [28], [29], [30] and deep learning methods [20], [26], [31]. The most widely prevalent filtering approach is based on the nearest neighbor (NNb) method and hence on spatiotemporal correlation [22], [23], [28]. In such filters, the properties of the previously generated events in a spatiotemporal neighborhood are utilized to determine a newly arrived event represents real activity. The parameters of the spatiotemporal window have to be tuned by the user. Fig. 2 shows the representation of event spatiotemporal neighborhood, where the newly arrived event data at t_i is marked as a red pixel and its spatial neighborhood is shown in blue. Therefore, such approaches require additional memory resources to retain the previous and the newly arrived events' properties for processing.

The BA filter proposed by Delbruck [28] classifies events that have less than eight other events in their spatiotemporal neighborhood as noise. One drawback of such an approach is observed when two BAs are close enough in one spatiotemporal region where the filter would consider them as real-activity events. Furthermore, Liu *et al.* [23] proposed a filter to tackle the problem of increased memory requirements by subsampling pixels into groups, where instead of projecting every pixel into a memory cell, one memory cell would hold a subsampled group of pixels. The filtration accuracy relies heavily on the subsampling factor, where the filtration accuracy decreases when the subsampling factor is greater than 2.

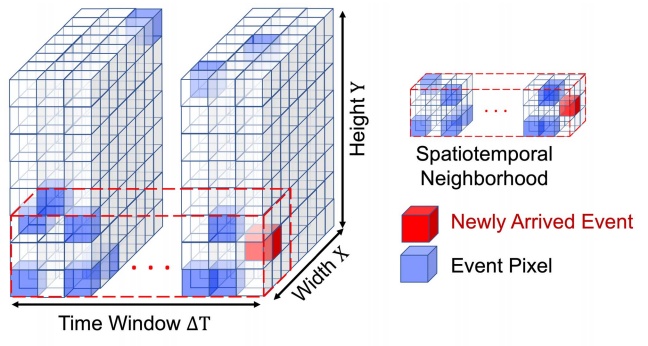


Fig. 2. Example of event spatiotemporal neighborhood.

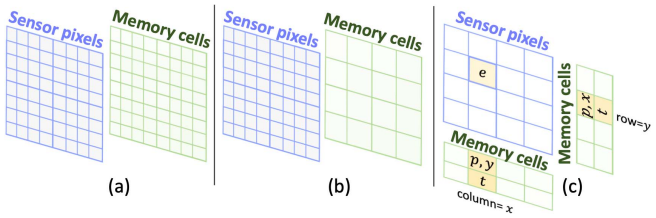


Fig. 3. Examples of memory strategy of different spatiotemporal filters [22]: (a) one memory cell per pixel [28], (b) one memory cell per two subsampling group [23], and (c) two memory cells for each column and row [22].

Khodamoradi and Kastner [22] proposed another storage technique for events and their timestamps to utilize less memory space. In particular, the most recent event in every row and column is stored along with its corresponding polarity and timestamp into two 32-bit memory cells. Hence, if two events are acquired in the same column, but two different rows, within a short temporal window, the recent event will override the old one in the memory. This is a serious limitation of this approach as establishing spatial correlation is deemed impossible, and thereby, more real-activity events could be sorted out as noise. Fig. 3 shows the techniques used to store events in the memory prior to filteration as proposed in [22], [23], and [28].

To overcome memory and computational complexity issues, Feng *et al.* [21] proposed a density matrix in which each arriving event is projected into its spatiotemporal region. The denoising process in this method consists of two steps: 1) removing random noises and 2) removing hot pixels (permanent active or permanent broken event pixels). Moving to the learning-based denoising approaches, in the literature, Baldwin *et al.* [20], [31] and Duan *et al.* [26] proposed a CNN and U-net network to filter DVS noises, respectively.

It is also evident that the performance of the existing denoising methods relies on tunable parameters, e.g., spatiotemporal window size, event camera settings, environmental illumination conditions, and camera motion dynamics [20], [21], [22], [26], [31]. Such parameters are application-dependent and manually tuning them may lead to satisfactory denoising results, especially in good lighting conditions. Despite setting the camera parameters to their optimal values though, features or edges of moving objects in very low illumination conditions are highly scattered and very noisy. In order to extract meaningful information from varying light conditions, the need for a method that can reject these noises and sharpen the real event data is essential. Nevertheless, spatiotemporal correlation-based and deep learning methods of ED remain largely unexplored.

B. GNNs and Transformers

GNNs are deep learning models that operate on non-Euclidean data structures such as graphs. GNNs consider the properties of each graph node and its connectivity within its neighborhood, regardless of the order in which data are provided to the neural network. It is also worth mentioning that the size of the input graph could be variable for the same network which makes GNN very well-suited for the application in hand. Due to its expressive power and model flexibility, GNN has recently been employed in a wide range of applications, e.g., visual understanding on images [32], [33]. Interested readers can explore more details in this direction in these recent surveys [34], [35].

There are different types of graph representations exhibiting various levels of complexity (i.e., number of connections and dimension) to address the problem in question. For instance, the work proposed in [36] and [37] designed graphs to represent point clouds and ground vehicle poses, respectively. The features of the nodes and edges in each graph encode the information necessary to perform the problem in hand, such as the point 3-D coordinates and the 2-D pose of the robot. In [36], a stack of EdgeConv layers is proposed to capture and exploit fine-grained geometric properties of point clouds that are then employed to carry out classification and segmentation for point cloud data. Another graph convolutional layer is proposed in [37], called PoseConv, to carry out global optimality verification of 2-D pose graph SLAM.

There are several types of GNNs, designed to fit different graph structures for different tasks. Our proposed algorithm adopts a message-passing algorithm on graphs, which is carried out in two stages: message passing and aggregation [34]. To construct a graph with a unique signature that reflects the nature of input data, in this work, spatiotemporal correlation functions are used. This is to reflect the nonlinear nature of the noise associated with DVS event streams. In addition, the graph isomorphism problem might occur when two different graphs might have an identical representation when reduced by the aggregation function. Inspired by [38], we employ a nonlinear activation within the aggregation stage to handle the graph isomorphism issue. This is to generate a unique graph signature to represent the spatiotemporal correlation between the nodes of the constructed graphs.

Recently, transformers have demonstrated the state-of-the-art performance on a multitude of applications, including natural language processing [18] and vision systems [16], [39], [40]. The self-attention head captures the relationship between inputs and outputs and supports parallel processing of sequential recurrent networks. In this article, we demonstrate the scalability of transformers on neuromorphic vision sensors and their capability to handle the asynchronous nature of events. This is designed within the graph layer that employs a message-passing algorithm to process the dynamic and variant nature of event streams. The output of the graph is then processed by the transformer, prior to the final classification stage that removes noise from the event stream.

III. Proposed Framework

In this article, a novel GNN-Transformer is proposed and trained to predict whether an incoming DVS event represents noise or a real log-intensity variation in the scene. Real log-intensity variation is a representation of a meaningful feature within the scene, e.g., the edge of an object. The overall framework of the proposed ED algorithm is shown in Fig. 4. In the following, we explain each component in detail.

A. Known-Object Ground-Truth Labeling

The availability of labeled datasets is key to the success of supervised learning algorithms. To that end, we propose a novel offline methodology, referred to as KoGTL, which classifies the DVS event stream into two main classes: real or noise event. We use KoGTL to generate labeled datasets and train a neural network to predict whether an event represents noise or real activity in the scene.

1) Experimental Setup: The main idea behind the KoGTL is to use a multitrial experimental approach to record event streams and then perform labeling. More specifically, a dynamic active pixel vision sensor (DAVIS346C) is mounted on a Universal Robot UR10 6-DOF arm [41], in a front forward position and repeatedly moved along a certain (identical) trajectory under various illumination conditions. The UR10 manipulator ensures a repeatability margin of 100 μs along a trajectory when performed repeatedly. The DAVIS346C provides a spatial resolution of 346 $\times$ 260, a minimum latency of 12 μs, a bandwidth of 12 MEvent/s, and a dynamic range of 120 dB [42]. The events are recorded along with two other measurements: 1) the camera pose at which the data were recorded, which we obtain through kinematics of the robot arm, and 2) the intensity measurements from the scene obtained using the augmented active pixel sensor, which is referred to as APS images hereafter.

Four experimental scenarios are adopted where data are acquired from the repeated transnational motion of the robot along square trajectories under different lighting conditions: particularly $\sim$750, $\sim$350, $\sim$5, and $\sim$0.15 lux. Streams of events with the corresponding APS images and robot poses were acquired for about 5 s per experimental scenario. Although the camera motion is identical in all experiments and the depicted scene (APS image) does not change, the properties of the event streams vary due to changes in illumination. Two of the experimental scenarios are used for training the proposed ED method, while the other two are used exclusively for testing and model evaluation.

2) Labeling Framework: The proposed KoGTL labeling algorithm is divided into three main stages, including event-image synchronization, event-edge fitting, and event labeling, as shown in Fig. 5.

a) Event-image synchronization: All the recorded experiments are first synchronized based on the time at which the robot arm has started moving [Fig. 5-(I)]. Consequently, following identical camera trajectories allows for synchronizing events and APS images across different lighting conditions. More specifically, events recorded under poor lighting conditions can be overlaid on APS images captured at the same camera pose under good lighting conditions, given that the scene is identical across all experiments. This facilitates matching events recorded in low-lighting conditions to alternative APS image features representing the same scene, which is extremely crucial for the success of the second stage. This would not have been possible using the APS images captured in low-lighting conditions where variations in intensities and hence features (edges) from the scene are absent.

b) Event-edge fitting: In the second stage, Canny edge detector [43] is used to extract edges from the APS images captured along the trajectory under good lighting conditions. The events captured between two consecutive APS images

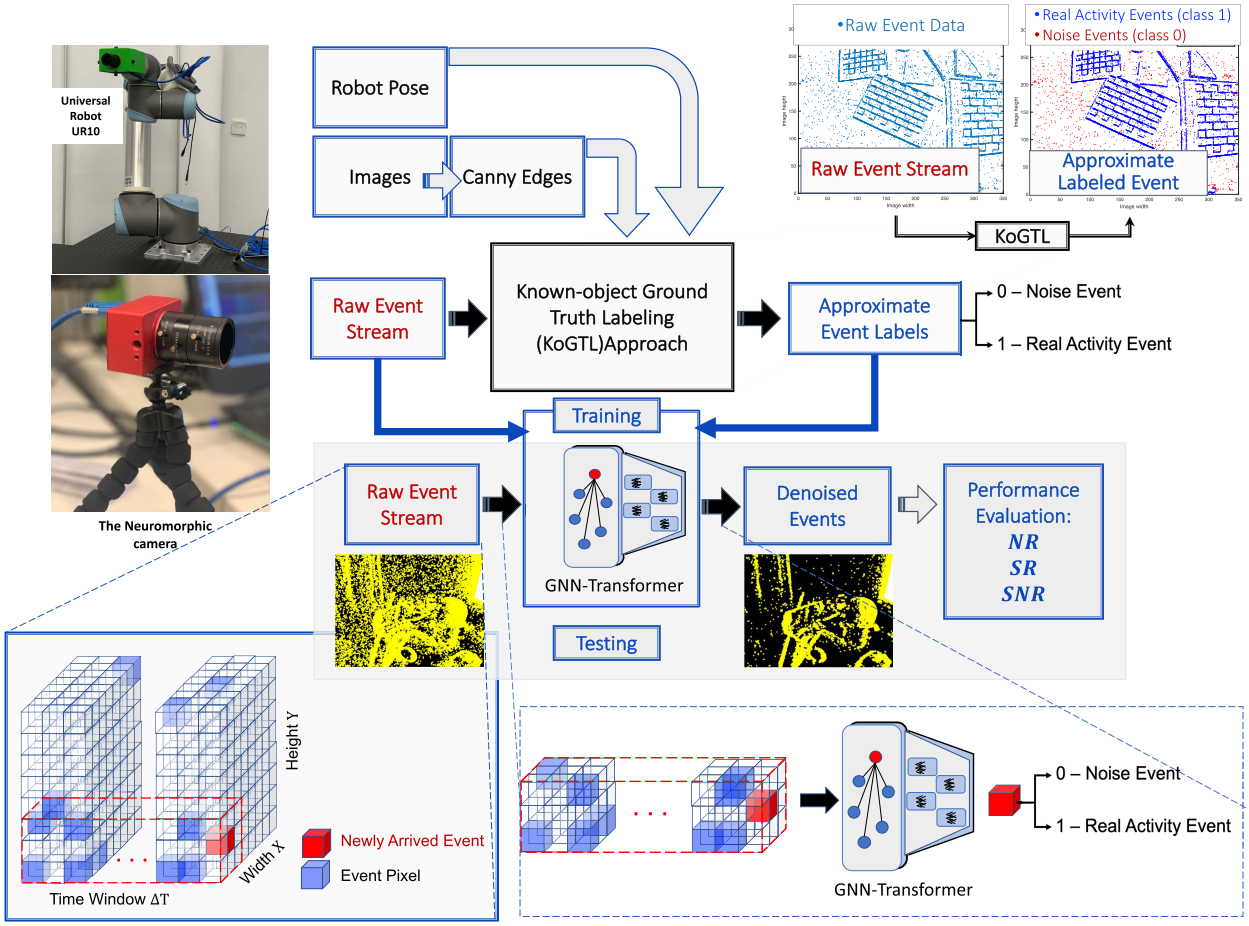


Fig. 4. Proposed ED framework. A GNN-Transformer-based ED algorithm is developed and trained on event datasets and generated and labeled using the proposed KoGTL approach. The proposed algorithm classifies incoming event streams into real-activity events or noise.

($t_{APS,i} <= t_{event} < t_{APS,i+1}$) are accumulated for every lighting scenario forming a 2-D vector, as shown in Fig. 5-(I). Using the iterative closest point (ICP) fitting technique [44], event data are fit to their corresponding APS edge data. Fitting was done in several stages because of the high temporal resolution of DVS data acquisition. Events might slightly deviate from APS edges due to imperfections in the time synchronization of events and APS data. Therefore, ICP is used to perfectly overlay them and correct any resulting spatial shift, as shown in Fig. 5-(II).

c) Event labeling: In the third stage, events that were fit to edges in the APS images are labeled as real-activity events (Class 1), as shown in Fig. 5-(III). Other events that fall out of a spatial window around edge pixels (between $+B$ and $-B$ pixels) are considered noise (Class 0). For our dataset, events are classified as noise when they are more than two pixels away (i.e., $B = 2$) from an edge in the APS image. This window size was selected based on visual observation of the fitting results using multiple B values.

B. Proposed GNN-Transformer Algorithm for ED

In this section, we explain the proposed GNN-Transformer for ED, as shown in Fig. 6. GNN-Transformer consists of three main stages: event graph construction, message passing on graphs, and event classification.

1) Event Graph Construction: Unlike conventional image frames, event data arrive asynchronously within a spatial resolution of $H \times W$ pixels (Fig. 6-I). Every pixel encodes log-intensity variations in the visual scene and is represented by a tuple $e =< x, y, t, p >$, where (x, y) are the pixel coordinates at which an event occurred, t is the event's timestamp, and p is the event's polarity (either 1 or -1, signifying an increase or a decrease in the intensity, respectively). A sequence of events within a spatiotemporal neighborhood is referred to as a local volume. The local volume is defined in terms of its spatial ($L \times L$) and temporal (T) dimensions around the event of interest. For example, if $L = 1$ and $T = 1$, the local volume includes the events arriving in a spatial window of 3×3 pixels around the event of interest in the previous 1 ms.

When a new event arrives, e_i (Fig. 6-II), a graph G that represents the local volume of the event is constructed (Fig. 6-III). The nodes of the graph are all the events in the defined local volume. Every node has three features $< (x_j), (y_j), (t_j) >$, where j is a node in the graph, x_j and y_j are the pixel coordinates at which the event occurred, and t_j is the event's timestamp. In this work, we omit the use of event polarity as a node feature because event polarity is affected by the sensitivity of events to changes in scene illumination, which may vary with different camera parameters. Directed edges are added from every node in the graph to the event of interest. More specifically, all neighboring events (nodes) will be connected to the newly arrived event (node or event of interest) that will be identified by the neural classifier. It is worth noting that the graph could be of variable size, i.e., every sample might include a different number of nodes. A very

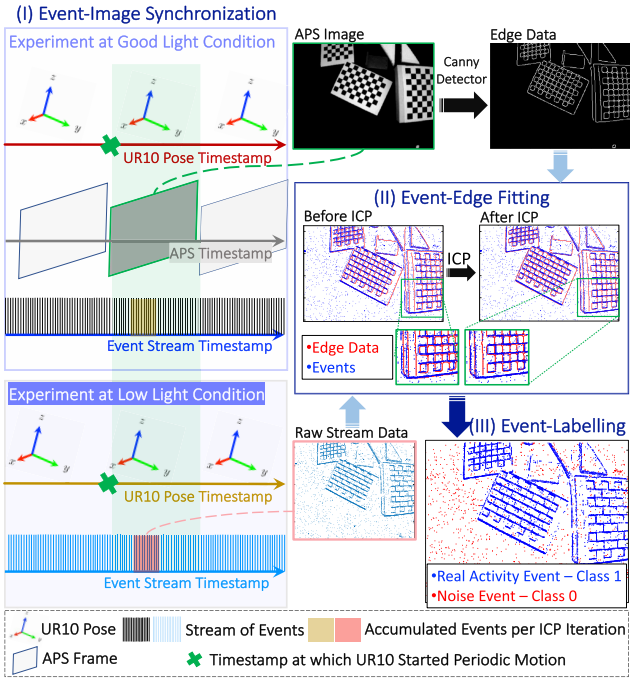


Fig. 5. KoGTL labeling framework. KoGTL is a novel DVS event labeling methodology developed to classify DVS events, acquired under various illumination conditions, into two main classes: real event or noise. The proposed KoGTL labels events that are acquired using a multitrial experimental approach, along with two measurements, camera pose and intensity measurements of the scene.

important property of GNNs is their ability to handle graphs of varying sizes, i.e., including variable number of nodes. This makes our approach more flexible since it facilitates the operation on events arriving asynchronously at a variable rate.

2) Message Passing on Graph—EventConv Layer: After constructing the event graph, messages are exchanged along the outgoing edges, from source nodes j to the node representing the newly arrived event i in the graph. The process of computing, sending, and aggregating the messages at the receiving node i is carried out by the proposed EventConv layer. Every node constructs a message consisting of its three features and sends it to node i for further processing. After receiving all the messages, node i, which represents the newly arriving event, processes and aggregates them. More specifically, the average of each of the node features $<(x), (y), (t)>$ across the graph is computed [Fig. 6-(1)]. The average values $\overline{x}$, $\overline{y}$, and $\overline{t}$ are then used to estimate the spatiotemporal correlations among the events in the event graph G. More specifically, the relationship between the event of interest and its neighboring events in space and time is encoded into seven quantities, which are (Q_1) the spatial difference in x, (Q_2) the spatial difference in y, (Q_3) the temporal difference, (Q_4) the standard deviation in x, (Q_5) the standard deviation in y, (Q_6) the standard deviation in t, and (Q_7) the euclidean distance. The computations of these quantities are shown in Fig. 6-(1) and denoted as $(Q_{1,L}, \ldots, Q_{7,L})$, where L represents the node index. These quantities were selected based on the results of an ablation study, as described in the following. Each of these quantities is passed through a linear layer followed by a sigmoid activation function prior to aggregation. Quantities of the same type across the received messages are summed up. This operation results in a 1-D vector representing a unique graph signature, which is referred to as h. The uniqueness of graph G signature circumvents the problem of isomorphism where two different graphs are represented by the same signature after being reduced in the aggregation stage [38]. Message passing and aggregation steps are carried out as part of the GNN, which is used in conjunction with transformers to perform classification. The steps explained above are shown in Fig. 6-(1).

3) Proposed GNN-Transformer Classifier: The overall architecture of the proposed learning-based classifier consists of two main parts, including a GNN and a transformer. In this section, more details about the structure selection are explained. Overall, for every acquired event in the stream, a graph is constructed to reflect the spatiotemporal correlations between this event and the previous events in its neighborhood. The proposed GNN operates on these graphs and outputs a graph signature, previously referred to as h. This graph signature is passed to the transformer for further processing. More particularly, the graph signature h is mapped to another representation by the transformer network, and finally, the binary classification is performed. The output of the proposed GNN-Transformer is a noise-free event stream that accurately resembles the activity in the scene.

Transformer is a sequence-to-sequence encoder–decoder network [17]. The self-attention mechanism encapsulates the interactions between all elements of a given sequence for structured prediction tasks. The attention mechanism with the query–key–value (QKV) model enables the transformer to have extremely long-term memory [17] and to execute dependencies between input and output and consequently execute more parallelization. The multihead attention layer comprises multiple stacks of self-attention. A multihead attention mechanism encapsulates a given sequence of elements into multiple jointly complex relationships by projecting them into three learnable weight matrices, called query, key, and value. In these matrices, computed weight distribution on the input sequence reflects the uniqueness of graph signature through assigning higher values to more representative elements. Basically, each element in a given input sequence in the multihead attention layer is updated by concatenating and aggregating global representative information.

Given a graph signature h with n elements $(h_1, h_2, \ldots, h_n)$, the objective of self-attention is to encode the global interaction information that exists among the elements. To achieve this, three learnable weight matrices are defined: queries $(W^Q \in R^{n \times d_q})$, keys $(W^K \in R^{n \times d_k})$, and values $(W^V \in R^{n \times d_v})$, where W is the learnable weight matrix, n is the size of the input features in h, and d_q, d_k, and d_v represent the dimensions of query, key, and value vectors, respectively, $d_q = d_k = d_v = n$ in our model. In the first step, the input sequence h is projected onto these weight matrices to obtain $Q = hW^Q$, $K = hW^K$, and $V = hW^V$. $Z \in R^{n \times d_v}$ is the output of self-attention layer and is computed as follows:

$$Z(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_q}}\right)V. \quad (1)$$

The most commonly used attention functions are the additive attention [45] and dot product attention [17]. In our model, dot-product attention, which is a simple matrix multiplication, is selected to update the state within the encoder and decoder units. This makes the attention process and its computations much faster and more space-efficient. In the multihead attention process, outputs from d self-attention units

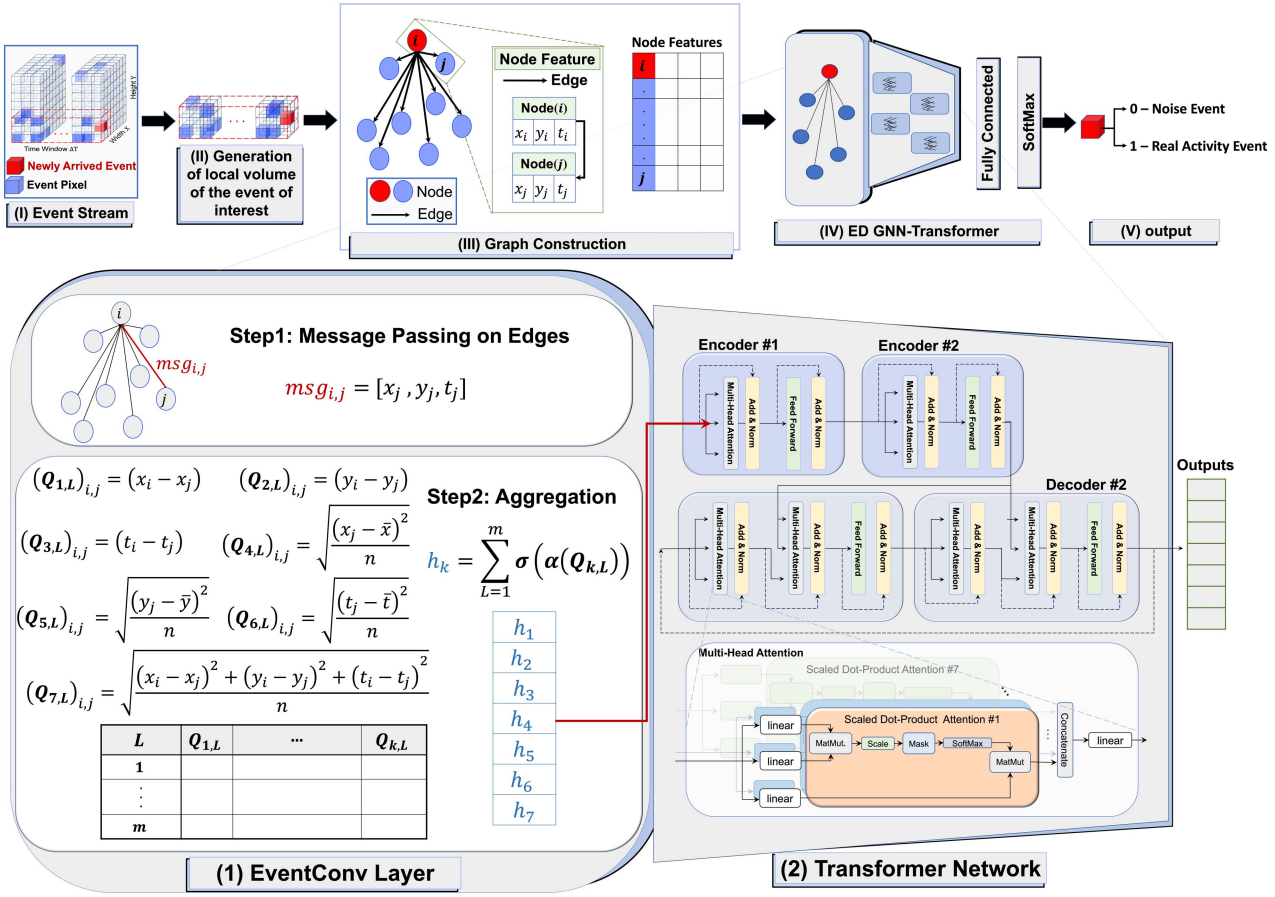


Fig. 6. Framework of our GNN-Transformer classifier for ED. Note: x and y are the pixel coordinates at which the event occurred. t is the event's timestamp. i and j are the source and destination nodes where a message is transferred in Step1-(1) EventConv layer, respectively. $Q_{1,L}, \ldots, Q_{7,L}$ are quantities that reflect spatiotemporal properties in the graph, where L represents the node index and m denotes the number of events in the local volume. h is the event graph signature. α is a learning parameter. σ is a sigmoid activation function.

are concatenated into one vector $[Z_1, Z_2, \ldots, Z_d]$ and are then projected by an output weight matrix $W^o \in R^{nd \times n}$ as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(Z_1, Z_2, \ldots, Z_d)W^o. \quad (2)$$

Furthermore, the multihead attention transformer facilitates identification of jointly complex relationships and makes the model easier to interpret.

a) Transformer encoder: The architecture of the encoder and decoder layers within the transformers follows the original structure in [17], which consists of a multihead self-attention (**MHA**) unit and a feedforward network **FFN**. The mathematical operations in a single encoder unit can be formulated as follows:

$$q_i = k_i = v_i = \mathbf{LN}(h_{i-1}) \quad (3)$$
$$y_{i-1} = h_{i-1} \quad (4)$$
$$y'_i = \mathbf{MHA}(q_i, k_i, v_i) + y_{i-1} \quad (5)$$
$$y_i = \mathbf{FFN}\big(\mathbf{LN}(y'_i)\big) + y'_i \quad i = 1, 2, \ldots, N \quad (6)$$
$$\left[F_{Ei}, F_{Ei+1}, \ldots, F_{EN}\right] = \left[y_i, y_{i+1}, \ldots, y_N\right] \quad (7)$$

where N denotes the number of encoder layers, **MHA** represents the multihead self-attention module, **LN** denotes the operation of layer normalization [46], and F_E denotes the output of the decoder layer. **FFN** is the feedforward network, which contains two fully connected layers with a ReLU activation function in between as in the following equation:

$$\mathbf{FFN}(x) = \max(0, x W_1 + b_1)W_2 + b_2. \quad (8)$$

b) Transformer decoder: For the transformer decoder unit, it takes the decoder's outputs as inputs and has two **MHA** modules followed by an **FFN**. The mathematical operations within a single decoder unit can be formulated as follows:

$$z_{i-1} = \left[F_{Ei}, F_{Ei+1}, \ldots, F_{El}\right] \quad (9)$$
$$q_i = k_i = v_i = \mathbf{LN}(z_{i-1}) \quad (10)$$
$$z'_i = \mathbf{MHA}(q_i, k_i, v_i) + z_{i-1} \quad (11)$$
$$q'_i = k'_i = v'_i = \mathbf{LN}(z_{i-1}) \quad (12)$$
$$z''_i = \mathbf{MHA}\left(q'_i, k'_i, v'_i\right) + z'_{i-1} \quad (13)$$
$$z_i = \mathbf{FFN}\big(\mathbf{LN}\left(z''_i\right)\big) + z''_i \quad i = 1, 2, \ldots, l \quad (14)$$
$$\left[F_{Di}, F_{Di+1}, \ldots, F_{Dl}\right] = \left[z_i, z_{i+1}, \ldots, z_l\right] \quad (15)$$

where l denotes the number of decoder layers and F_D represents the output of the transformer unit ($F_D \in R^{n \times 1}$), which reveals the important features to uniquely represent the graph signature (h).

The output of the coupled GNN-Transformer is finally passed to a fully connected layer that generates a 2×1 tensor

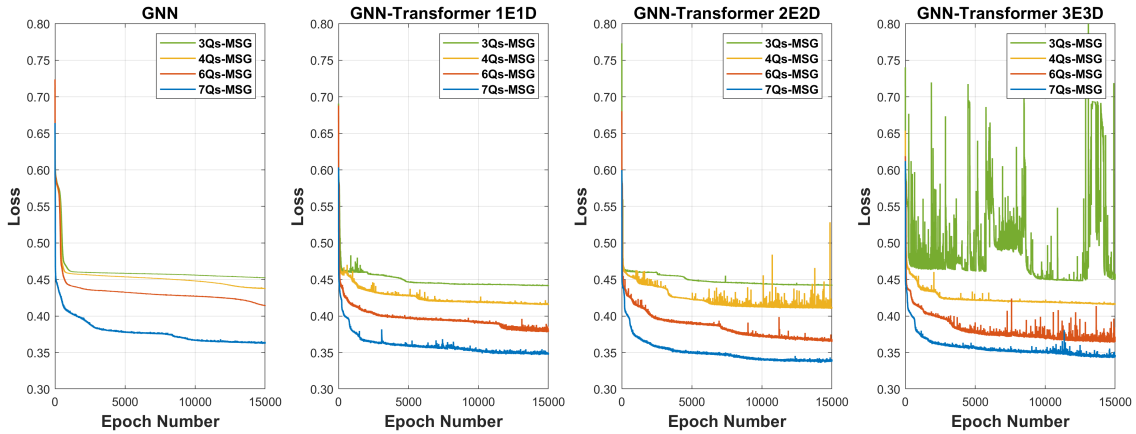


Fig. 7. Ablation study results—loss curves obtained upon training various network architectures as part of the automated search for the best suited neural network architecture.

for every sample in the dataset, where 2 is the number of classes: real log-intensity change or noise. The output tensor is passed to a softmax function (16), where it is rescaled so that the elements are in the range [0, 1] and sum up to unity. The rescaled elements represent the probabilities that the event under investigation represents noise or real activity

$$\textbf{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{2} e^{x_j}}. \tag{16}$$

Supervised learning is performed using the backpropagation algorithm to train the GNN-Transformer network. Pytorch [47] implementation is used for constructing all the neural networks and performing training and testing. The training process is carried out to minimize the cross-entropy loss function using the Adam optimizer [48] with a learning rate of 0.001.

c) Ablation study: To select the most suited structures of both the GNN and the transformer, an automated search routine was developed. The automated search routine spanned several parameters, including the graph structure, the message operation, the aggregation functions, the number of EventConv layers in the GNN, the activation functions, and the number of encoder–decoder units in the transformer. Such parameters reflect the nonlinear capacity of the model and hence need to be carefully selected to best suit the problem in question. It was observed that several architectures have achieved comparable performance and were able to correctly classify the majority of real-activity and noise events.

Fig. 7 reports the loss obtained by the highest performing architectures on the training dataset among the tested neural networks. The loss curves are grouped based on the adopted neural network architecture: GNN, GNN in conjunction with a transformer of a single encoder–decoder layer (GNN-Transformer 1E1D), GNN in conjunction with a transformer of a double encoder–decoder layer (GNN-Transformer 2E2D), and GNN in conjunction with a transformer of a triple encode–decoder layer (GNN-Transformer 3E3D). For every architecture, the number of quantities composing the messages that characterize the spatiotemporal correlation within the graph was varied. More specifically, four combinations of quantities in the message were tested as indicated in the following.

1) **3Qs-MSG:** Q_1, Q_2, and Q_3.
2) **4Qs-MSG:** $Q1$, $Q2$, $Q3$, and $Q7$.
3) **6Qs-MSG:** $Q1$, $Q2$, $Q3$, $Q4$, $Q5$, and $Q6$.
4) **7Qs-MSG:** $Q1$, $Q2$, $Q3$, $Q4$, $Q5$, $Q6$, and $Q7$.

The performance of all the attempted networks is evaluated using unseen testing datasets, which are composed of streams of events obtained experimentally. The performance evaluation metrics used to compare the training and validation results are the accuracy, signal ratio (SR), noise ratio (NR), and signal-to-noise ratio (SNR) as computed with respect to the ground-truth labels obtained using our proposed KoGTL for each event.

The training and testing results have proven that the GNN-Transformer architecture with 7Qs-MSG in the EventConv layer as described in Section III-B2 and a transformer with a double encoder–decoder layer showed the best performance among all candidate neural classifiers in terms of the noise filtration accuracy as reported in Table IV (see the Appendix). The proposed GNN-Transformer architecture is shown in Fig. 6-IV.

It is worth noting that the quantities included in the messages play a pivotal role in reflecting the spatiotemporal correlation of the event and its neighboring events and thus in the overall performance of the filter as clearly shown in loss curves of the GNN-Transformer 3E3D. More specifically, although the architecture of the neural network was complex enough, the number of quantities in the message drastically affected the filter's performance.

IV. Experimental Evaluations

The proposed GNN-Transformer algorithm for ED is tested qualitatively and quantitatively in multiple scenarios to demonstrate its validity, effectiveness, and generalization. The training process, including training and testing data preparation, is described in Section IV-A. In Section IV-B, the evaluation metrics used to quantify the results are presented. Section IV-C presents the quantitative performance analyses of the developed GNN-Transformer model. Moreover, the GNN-Transformer model is benchmarked against other existing ED methods, where the developed model's capability, effectiveness, and validity are discussed. In addition, the performance of the model is evaluated qualitatively on part of the datasets that we have recorded but have not exposed to the network during training, as well as several publicly available datasets as presented in Section IV-D. This is to prove the model's generality and robustness to various illumination conditions and unseen data.

A. Training and Testing Datasets

They are constructed from experiments recorded in our lab as well as other publicly available datasets. Training is exclusively done using our recorded dataset because of the availability of ground-truth labels to support supervised learning. Testing, on the other hand, is done on both recorded and publicly available datasets where quantitative and qualitative evaluations are done.

Recorded experiments were conducted following the approach described in Section III-A using the iniVation's DAVIS346C DVS [42]. Four lighting conditions were used to record experiments; very good lighting (∼750 lux), office lighting (∼300 lux), low light condition (∼5 lux), and moon light condition (∼0.15 lux). Every experimental scenario includes scenes recorded when the camera is static or is starting translational motion, and scenes recorded when the camera is moving in four different directions. In the former case, static noise pixels can be detected and learned accordingly. The latter cases exhibit the dynamic nonlinear nature of event and noise generation as well as spatiotemporal correlations of an event and its neighborhood when the camera is in motion.

Samples from the experiments recorded under very good lighting (∼750 lux) and low light conditions (∼5 lux) were used for quantitative analysis (training and testing). Each sample consists of a newly arrived event and its corresponding neighboring events within the defined spatial and temporal window. More specifically, for each scenario, a total of 8000 samples were randomly selected from each of the five scenes (static and motion in four directions): 4000 real-activity events and 4000 noise samples. This is to ensure that the training dataset is balanced and is not biased toward one class more than the other. Hence, a total of 80k samples constitute the dataset, 80% of which are used for training and 20% are used for testing.

Moreover, a qualitative analysis of the model's performance on two recorded experiments (∼300 and ∼0.15 lux) and 11 publicly available datasets was carried out. The publicly available datasets [12] include indoor and outdoor scenarios and were recorded at numerous illumination conditions and using different motion dynamics, as summarized in Table I.

Prior to training the model, every sample event and its corresponding neighborhood are used to construct a graph, which is used as the input to the GNN. The size of the neighborhood, i.e., the local volume, is selected to be a maximum of ten nodes (or events) within 5×5 pixels window centered at the event of interest in the proceeding 50 ms. In case more events were acquired in this volume, only the latest ten are included in the graph. It is worth mentioning that the volume size was selected after several experiments with varying volume parameters. It was observed that ten neighboring events in the local volume are sufficient to delineate the spatiotemporal correlations and hence make a decision on whether the event of interest is real or noise.

To expedite training and convergence, it is common practice to normalize all the inputs to the neural network to a common range. In this work, all inputs are rescaled to the range $[0.05, 0.95]$, excluding values very close to 0 and 1 to avoid the issue of neuron saturation, which causes the problem of vanishing gradients. For example, the minimum and maximum values of sigmoid are 0 and 1, respectively. The corresponding derivative at those values drops to zero, causing gradients to vanish.

TABLE I

DESCRIPTION OF THE PUBLICLY AVAILABLE DATASETS USED FROM [12]

Name	Scene Description	Light Condition (lux)
***Simple*-Scene**	Simple 6DOF camera motions looking at simple objects and scenes with vibrant colors.	
SimpleFruit	Colorful fruits, fluorescent and window lighting.	1000
SimpleObjects	Colorful everyday objects, fluorescent and window lighting.	1000
SimpleObjectsDynamic	Colorful everyday objects being picked up, fluorescent and window lighting.	1000
SimpleWires1	Colorful rolls of wire, fluorescent and window lighting.	400
***Indoors*-Scene**	Natural indoor scenes including office, kitchen, rooms and corridors.	
IndoorsCorridor	Walking down dimly lit corridor, into room with bright windows.	80-1000
IndoorsDark25ms	Desk illuminated by two monitors, exposure set to 25ms.	2
IndoorsFootball1	Foosball table, fluorescent lighting.	200
IndoorsKitchen1	People in kitchen, fluorescent lighting.	200
***Driving*-Scene**	Footage from front windshield of car driving around country, suburban and city landscapes. Features tunnels, traffic lights,vehicles and pedestrians during the day in sunny conditions.	
DrivingCity4	Driving around the city,features tunnel and light traffic.	200-100,000
DrivingTunnel	Driving into long tunnel (15 seconds) and out into bright sunlight.	200-100,000
DrivingTunnelSun	10 second tunnel followed by direct sun in field of view.	200-100,000

B. Evaluation Metrics

To quantitatively evaluate the performance of the proposed denoising model and compare to state-of-the-art models on training and testing datasets, four evaluation metrics are used: accuracy, SR, NR, and SNR.

d) Accuracy: This metric measures the model's ability to correctly predict real-activity events and noise, as defined in the following equation:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{17}$$

where TP, FP, TN, and FN are the number of true positives, false positives, true negatives, and false negatives pixels, respectively. TP indicates the number of events that are correctly predicted as real-activity events, whereas TN indicates the number of events that are correctly predicted as noise.

e) Signal ratio: This metric represents the proportion of correctly predicted real-activity events with respect to the total number of real-activity events in the scene, which is also known as precision, as defined in the following equation:

$$\text{SR} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{18}$$

f) Noise ratio: This metric represents the proportion of incorrectly predicted noise events with respect to the total number of noise events in the scene, which is also known as the false omission rate, as defined in the following equation:

$$\text{NR} = \frac{\text{FN}}{\text{TN} + \text{FN}}. \tag{19}$$

g) Signal-to-noise ratio: This metric is the ratio of the number of correctly predicted real-activity events to the number of noise events incorrectly labeled as real-activity events as described in the following equation:

$$\text{SNR} = \frac{\text{TP}}{\text{FN}}. \tag{20}$$

The performance of the denoising model is considered better with higher SR and SNR values and lower NR values.

C. Quantitative Results

1) Evaluation on Training and Testing Datasets: In this section, the performance of the proposed GNN-Transformer-based ED model is compared against state-of-the-art denoising methods, namely, EDnCNN [20], Yang filter [21], Khodamoradi filter [22], Liu filters [23], and NNb filter [24]. All filters are tested on the same dataset, which was used to train our proposed approach. The dataset was randomly split into training and testing subsets, where 80% of the samples were used for training and 20% were used for testing (not exposed to the network during training).

EDnCNN filter's parameters were set to those mentioned in their published trained model that consists of 3 × 3 convolutional layers followed by two fully connected layers. To filter an event, a spatiotemporal window of 25 × 25 × 5 s centered at that event pixel is considered to construct the input feature to the model. More specifically, a 25 × 25 × k × 2 matrix is populated with the k most recent positive and negative events that were received prior to the event of interest, where k was set to 2. The pretrained EDnCNN model parameters [20] were used to perform accuracy evaluations on both our training and testing datasets. Yang filter's parameters were set to the default values reported in [21]. More specifically, the time window was set to 5 ms, the spatial window is 5 × 5 pixels, and the density is 3. As for the Khodamoradi filter, the time window was set to 1 ms, as in [21] and [22]. Two downsampling factors S of Liu's filter were used $S = 1$ and 2 where the timestamp of 2 × 2 and 4 × 4 pixels was stored in one memory cell and the time window was set to 1 ms, as tested in [21]. The working principle of Liu and Khodamoradi filters was previously mentioned in Section II (see Fig. 3(b) and (c), respectively). Finally, for the NNb filter, the size of the event's local volume is set to 3 × 3 pixels for 1 ms, as reported in their work [24]. The performance of these denoising methods was compared to that of the proposed GNN-Transformer approach as presented next.

Table II reports the filtration accuracy achieved by the GNN-Transformer network, EDnCNN filter, Yang filter, Khodamoradi filter, Liu filter, and NNb filter when evaluated on the training and testing datasets. It is worth mentioning that the training and testing datasets have equal numbers of real and noise events (50% real events and 50% noise events). It is observed that the GNN-Transformer outperforms all the other alternatives in terms of filtration accuracy. The proposed model has outperformed EDnCNN by 10.6% on the training dataset and 8.4% on the testing dataset. It has also achieved 12% higher training and testing accuracy compared to the Yang filter. The Yang filter has shown the best performance compared to other conventional filters (Khodamoradi, Liu, and NNb filters) in terms of filtration accuracy.

A high SNR value does not necessarily mean that a filter's performance is better than others. Rather, a high SNR value, a high SR value, and a low NR value together would indicate a good filtering performance. A clear example is the Khodamoradi filter, which achieved the highest SR (99%) and the highest NR (92%) values among other filters. These values mean that all input data have been considered real activity and no noise filtration took place. In other words, the filter could not distinguish between the incoming real-activity events and the accompanying noise.

Another example is Liu's filter, which achieved the lowest NR (1-2%) and a relatively low SR (10-30%). In this case, most of the input data have been considered as noise. This implies the weak denoising capability of Liu's filter. Meaningful real-activity events have been filtered out, and consequently, scene perception algorithms would fail to operate as expected.

To conclude, the best ED model is expected to have a high accuracy, SR, and SNR, and a low NR. Thus, our proposed GNN-Transformer has clearly outperformed all alternative filters and proved its capability to generalize to unseen datasets. Table II compares the number of correctly and incorrectly predicted real-activity events from the training and testing datasets.

TABLE II

PERFORMANCE OF THE GNN-TRANSFORMER CLASSIFIER COMPARED TO STATE-OF-THE-ART DENOISING METHODS ON THE TRAINING AND TESTING DATASETS

Training Dataset					
Event Denoising Method	TP	FP	TN	FN	Filtration *Accuracy*
Yang Filter [21]	15529	16471	29012	2988	69.60%
Khodamoradi Filter [22]	31889	111	2526	29474	53.77%
Liu Filter [23] (SubGroup by 2)	3665	28335	31225	775	54.52%
Liu Filter [23] (SubGroup by 4)	10149	21851	28429	3571	60.28%
NNb Filter [24]	7594	24406	30313	1687	59.23%
EdnCNN [20]	18830	13170	27082	4918	71.73%
GNN-Transformer (ours)	27012	4988	25684	6316	**82.34%**
Testing Dataset					
Event Denoising Method	TP	FP	TN	FN	Filtration *Accuracy*
Yang Filter [21]	3831	4169	7220	780	69.07%
Khodamoradi Filter [22]	7977	23	670	7330	54.04%
Liu Filter [23] (SubGroup by 2)	925	7075	7829	171	54.71%
Liu Filter [23] (SubGroup by 4)	2451	5549	7092	908	59.64%
NNb Filter [24]	1889	6111	7564	436	59.08%
EdnCNN [20]	4722	3278	6790	1210	71.95%
GNN-Transformer (ours)	6403	1597	6513	1487	**80.73%**

2) Evaluation on Our Recorded Dataset—Continuous Stream of Events: In this section, the proposed model is tested online on a continuous stream of events and then compared to state-of-the-art denoising techniques. In other words, instead of randomly selecting samples from the recorded experiments, the full stream of events generated by DVS is passed through each filter, which is then evaluated, as per our labeled dataset.

Filtering techniques were tested in two scenarios; the experiments recorded at ∼750 and ∼5 lux. In the first scenario, filtering was done over 600 ms, where SR and NR were evaluated every 10 ms, as shown in Fig. 8(a). The second scenario was run for 170 ms and the evaluation was done at 5-ms intervals, as shown in Fig. 8(b). Evaluations of SR, NR, and SNR over the full period of time for both scenarios are shown in Fig. 9(a) and (b). The total number of events included in this test is 7M and 0.1M for the first and second scenarios, respectively.

It is evident, through the conducted tests, that our proposed GNN-Transformer-based ED technique has achieved the best filtering performance compared to all the other filters. This proves the effectiveness of the proposed ED approach and shows robustness to different camera motion dynamics under

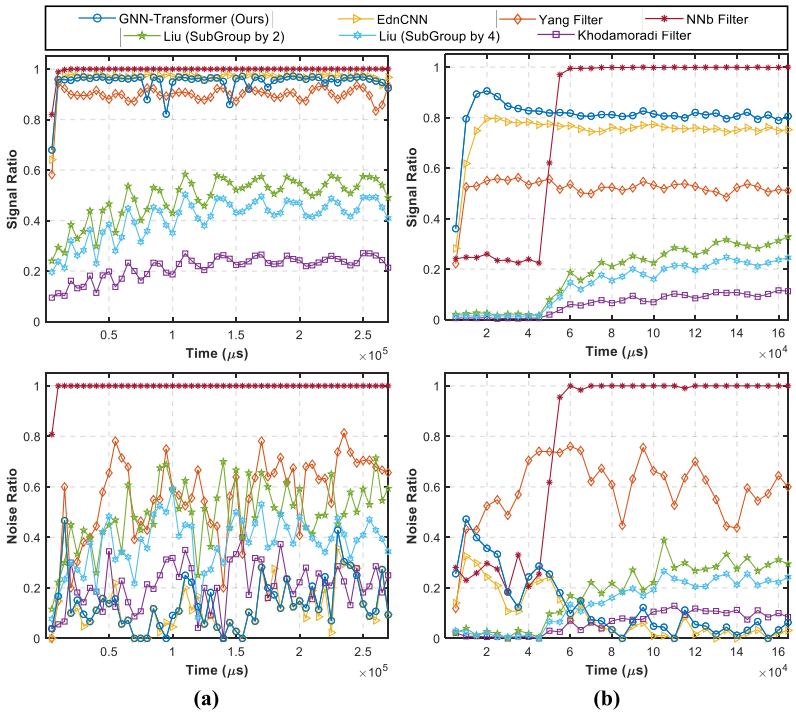


Fig. 8. SR, NR, and SNR ED performances of the GNN-Transformer model and state-of-the-art denoising methods—using sample stream of events recorded at (a) ∼750 and (b) ∼5 lux.

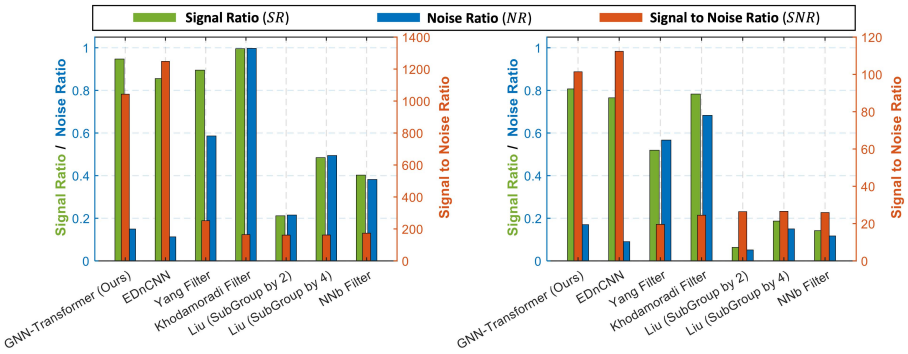


Fig. 9. SR, NR, and SNR ED performances of the GNN-Transformer model and state-of-the-art denoising methods—using sample stream of events recorded at (a) ∼750 and (b) ∼5 lux. The performance of the denoising model is considered better with higher SR and SNR values and lower than NR values. It can be observed that the best performing denoising methods are ours and EDnCNN [20]. However, for fair comparison and for these results to make sense, the metrics have to be analyzed collectively. It was observed that EDnCNN has considered a large number of events as noise, which decreased the NR value compared to ours. However, a significant amount of these filtered events belongs to meaningful features, i.e., were incorrectly labeled as noise, which resulted in a lower SR value than ours.

illumination variations. According to our evaluations, the second-best learning-based ED technique is the EDnCNN [20] filter and the best conventional ED filter is Yang filter [21]. Thus, further qualitative performance assessments of our proposed approach are conducted against those two filters only as presented in Section IV-D.

3) Computational Time Complexity and Memory Analysis: In this section, time and memory analyses of the proposed approach will be discussed and compared to the EDnCNN filter since both are based on using neural networks. A set of 10 000 event samples was selected from the *stairs* dataset presented in [20] to conduct the timing analysis.

The computational time analysis of the proposed algorithm was carried out on an ASUS laptop with Intel core $i7-7700HQ$@2.80 GHz $\times$ 4, NIVIDIA GeForce GTX 1050 Ti 4 GB. The analysis was done with and without GPU support in two modes: sequential mode—events were passed to the filter successively, one after the other, and batch mode—all events were passed to the filter as a single batch. The time needed to filter the events in each mode was recorded for both filters, as listed in Table III. In all cases, the time needed to complete the filtration was shorter using our proposed approach compared to EDnCNN. However, our approach achieved a large speedup of up to two orders of magnitude in the batch mode compared to the other filter when run on CPU and a speedup of up to one order of magnitude when run on GPU. This speedup is significant as operation in the batch mode is certainly necessary due to the high temporal resolution of the event camera and due to the working principle of the event camera that enables 346 $\times$ 260 pixels to be active simultaneously. In other words, the proposed approach is capable of handling batches

TABLE III

TIME IN SECONDS TO FILTER EVENTS USING OUR PROPOSED APPROACH AND EDNCNN METHOD [20]. NOTE THAT μ AND σ REPRESENT THE MEAN AND STANDARD DEVIATION, RESPECTIVELY

Processing Unit	Event Denoising Model	Sequential-mode $\mu \pm \sigma$ (sec)	Batch-mode μ (sec)
CPU	EDnCNN [20]	$1.46 \times 10^{-2} \pm 2.38 \times 10^{-3}$	1.54×10^{-3}
	GNN-Transformer	$1.28 \times 10^{-3} \pm 2.07 \times 10^{-4}$	5.24×10^{-5}
GPU	EDnCNN [20]	$7.19 \times 10^{-3} \pm 3.12 \times 10^{-3}$	3.01×10^{-4}
	GNN-Transformer	$1.69 \times 10^{-3} \pm 2.65 \times 10^{-4}$	6.77×10^{-5}

of events concurrently in a very short period of time and hence preserves the high temporal resolution of the sensor. It is also worth noting that the proposed approach exhibited the fastest performance when processing events in a batch mode on a CPU, which obviates the need for sophisticated hardware to achieve fast and accurate noise filtration. This makes the proposed approach suitable for limited computational power and resource-constrained platforms such as high-speed unmanned aerial vehicle (UAV) control [49], UAV navigation [50], and space applications [51].

To project this analysis on a real-world scenario, consider the application of autonomous car driving where neuromorphic vision could be employed to observe the environment during navigation. As the speed of the vehicle increases, the number of generated events will proportionally increase resulting in a tremendous amount of events for processing. Faster processing of visual observations will thus result in a faster response to changes in the vehicle's surroundings. This will definitely reduce the probability of collisions and will enhance the effectiveness of the overall system.

The overall memory requirement per event classification is $5 \times 5 \times N_g$, where N_g is the number of events per graph and could range from 1 to 10 events, whereas in EDnCNN, the size of the input feature is $25 \times 25 \times 2 \times 2$. This clearly shows that our approach is more memory efficient than EDnCNN, where in case the graph in our approach had ten nodes (which is the maximum number of nodes per graph), the memory requirements are ten times less than that of EDnCNN.

D. Qualitative Results

In this section, two experiments from our recorded dataset, particularly those recorded at ∼300 and ∼0.15 lux, are used to qualitatively analyze the denoising performance of the proposed model against and EDnCNN and Yang filters. Sample filtering results, superimposed on APS images for better visualization, are shown in Fig. 10. The results clearly show that our model has filtered out most of the BA noise and maintained events representing relative motion of meaningful features in the scene as in Fig. 10(a). Although more scattered noise is present under low-lighting conditions as shown in Fig. 10(b), our proposed model was able to preserve the events that represent meaningful features (edges) in the scene. Conversely, the Yang filter has eliminated the majority of real-activity events from the scene while leaving some scattered ones that could be hard to interpret as edges or meaningful features. This proves the robustness of our model against illumination variations.

To further prove the validity and generalization of our proposed model, we have extensively tested it and compared it against others using 11 publicly available datasets. These recorded data were acquired from different camera motion

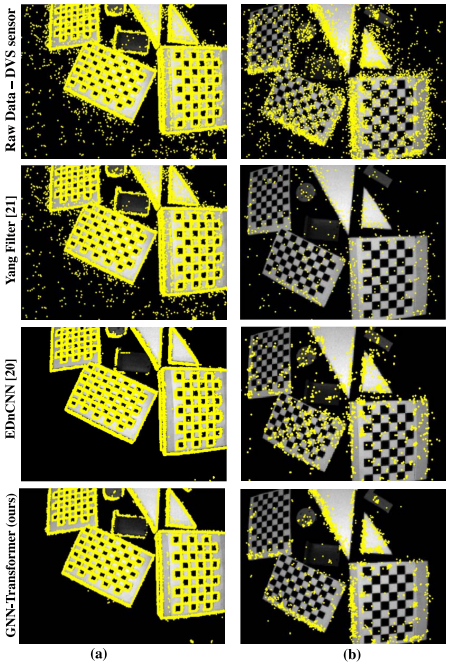


Fig. 10. Denoising results tested on our dataset (unseen data), denoised events from DVS (yellow dots) overlaid on the corresponding APS image. (a) Our dataset: Exp at ∼300-lux office light condition. (b) Our dataset: Exp at ∼0.15-lux moon light condition.

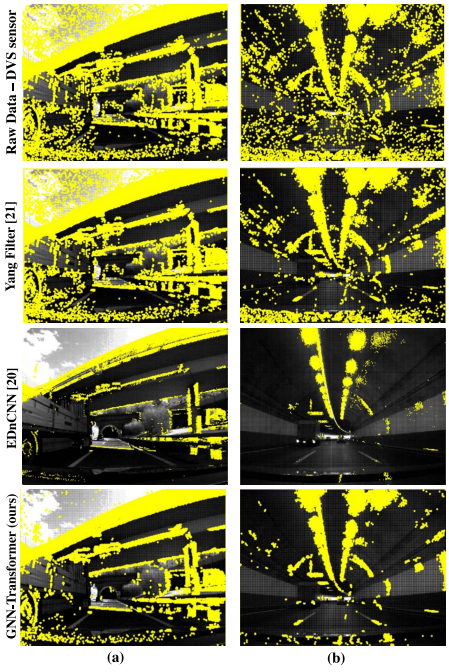


Fig. 11. Sample of denoising results tested on published datasets (unseen data), denoised events from DVS (yellow dots) overlaid on the corresponding APS image. (a) Published datasets: DrivingTunnel. (b) Published datasets: DrivingTunnelSun.

dynamics (type of motion and speed) and under different lighting conditions. Fig. 11 shows two examples of denoised events obtained using the proposed model, EDnCNN, and Yang filter. It was noticed that EDnCNN eliminated a large amount of events that belong to meaningful features in the scene. For instance, the filtered event stream corresponding

to the scene taken from the DrivingTunnelSun dataset shown in Fig. 11(a) lacks significant events that represent clear intensity variations as per the corresponding APS images. Such events were classified as noise using the EDnCNN filter. The same observation can be seen in the scenes from the other datasets such as DrivingCity4 in Fig. 11. Yang filter passes the majority of the events (both real and noise signals), thus making it more difficult to identify objects (edges) in the scene compared to our proposed model. Therefore, the GNN-Transformer-based ED model generalizes well to new scenarios under various illumination conditions without any further tuning of its parameters. More results are shown in Fig. 12 (see the Appendix), additional results document in <https://github.com/Yusra-alkendi/ED-KoGTL> and video <https://youtu.be/ZM76UaxbuJE>, which visualize the denoising performance of GNN-Transformer classifier compared to Yang filter [21] and EDnCNN [20].

V. Conclusion

In this work, we developed a novel algorithm to filter out the noise associated with event streams acquired by DVSs. The GNN-Transformer-based ED algorithm exploits the spatiotemporal correlations between events in a particular neighborhood to decide whether an incoming event represents noise or a log-intensity variation in the observed scene. To train the proposed GNN-Transformer model, a novel offline event labeling technique, KoGTL, is proposed to distinguish between noise and real events in event streams recorded under challenging lighting conditions. The labeled DVS data is made available to the public research community for benchmarking purposes. The proposed algorithm successfully operates on event streams irrespective of camera parameters, illumination conditions, and motion dynamics. This is attributed to the fact that the adopted graph structure of the input data preserves the spatiotemporal correlation between the events, rather than the raw properties of the events, solely. Such operation is carried out in the proposed EventConv layer. The proposed algorithm also operates on event graphs of variable sizes and thus handles the asynchronous nature of event streams.

Through extensive training and testing, the proposed algorithm has proven to achieve significantly high denoising performance under challenging illumination conditions. Our model is also tested on 11 publicly available datasets that were not exposed to the network during training. The model is able to successfully denoise the event streams, despite the fact that the data are recorded under conditions different than those of the training data, including different environmental conditions, various camera motions, and camera parameters. The quantitative results have demonstrated the denoising capability of the proposed algorithm with at least 8.8% higher filtration accuracy on testing sets compared to existing methods. Qualitatively, the results achieved by the proposed model have verified its effectiveness and generalization to previously unseen event graph data, irrespective of their sizes. This work has unveiled the power and potential of GNNs and transformers on event cameras.

In the future, we plan to demonstrate the significance of our proposed denoising approach by integrating it into other event-based computer vision algorithms such as motion segmentation, object detection, object tracking, and object recognition, under challenging lighting conditions. We also plan to exploit the potential of GNNs and transformers for other event-based vision algorithms. Another possible extension of the current work could be by integrating the denoising module together with vision algorithms and employing them for robot navigation purposes, autonomous driving cars [52], and healthcare applications such as human fall detection [53]. Eliminating noise events from the observed scene in such scenarios is foreseen to improve the accuracy of the vision algorithms responsible for localizing obstacles and detecting human fall accidents. Noise events, if not eliminated, may be mistaken for real changes in the scene intensities, which could result in false positive detections. In the case of autonomous driving, falsely detecting an obstacle along the way will interrupt the vehicle's trajectory and may cause it to take longer paths and more time, which is undesirable. As for human fall detection, noise events may decrease the accuracy of localizing a human and estimating the temporal window for the accident by inflicting erroneous information into the observation. To that end, integrating the proposed denoising method into such systems is envisioned to enhance their accuracy and effectiveness.

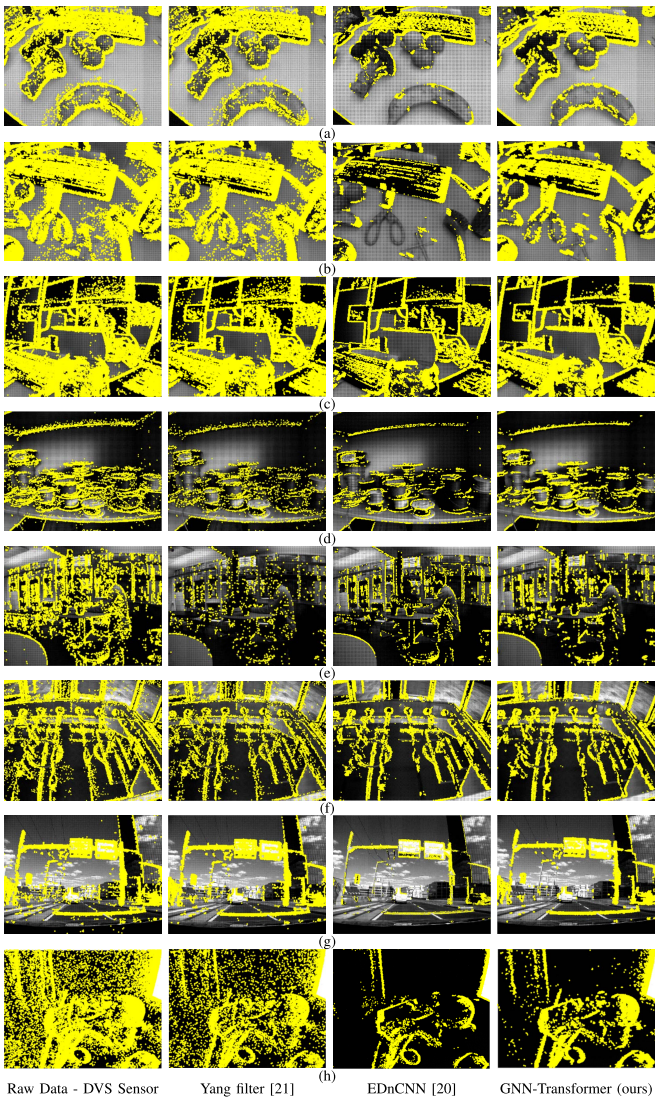


Fig. 12. Additional qualitative denoising results tested on the published dataset (unseen data), denoised events from DVS (yellow dots) overlaid on APS image.

TABLE IV

PERFORMANCE COMPARISON OF THE PROPOSED ED CLASSIFIER AND ITS NETWORK VARIANTS ON THE TRAINING AND TESTING DATASETS. NOTE THAT CASES I–IV DENOTE GNN, GNN-TRANSFORMER 1E1D, GNN-TRANSFORMER 2E2D, AND GNN-TRANSFORMER 3E3D, RESPECTIVELY

Training (Testing) Dataset					
Event Denoising Model	TP	FP	TN	FN	Filtration *Accuracy*
Case I - 3Qs-MSG	25750 (5702)	6250 (2298)	23905 (5862)	8095 (2138)	77.59% (72.28%)
Case I - 4Qs-MSG	26008 (5690)	5992 (2310)	24046 (6021)	7954 (1979)	78.21% (73.19%)
Case I - 6Qs-MSG	25269 (5315)	6731 (2685)	25111 (6213)	6889 (1787)	78.72% (72.05%)
Case I - 7Qs-MSG	22446 (5447)	9554 (2553)	27329 (6728)	4671 (1272)	77.77% (76.09%)
Case II - 3Qs-MSG	25745 (5659)	6255 (2341)	23998 (5911)	8002 (2089)	77.72% (72.31%)
Case II - 4Qs-MSG	31353 (7843)	647 (157)	12016 (2939)	19984 (5061)	67.76% (67.39%)
Case II - 6Qs-MSG	21420 (5394)	10580 (2606)	27093 (6702)	4907 (1298)	75.80% (75.60%)
Case II - 7Qs-MSG	21983 (5497)	10017 (2503)	27343 (6742)	4657 (1258)	77.07% (76.49%)
Case III - 3Qs-MSG	26879 (5731)	5121 (2269)	24099 (5938)	7901 (2062)	79.65% (72.93%)
Case III - 4Qs-MSG	24860 (5167)	7140 (2833)	26311 (6556)	5689 (1444)	79.95% (73.27%)
Case III - 6Qs-MSG	5168 (1229)	26832 (6771)	30972 (7721)	1028 (279)	56.47% (55.94%)
Case III - 7Qs-MSG	27012 (6403)	4988 (1597)	25684 (6513)	6316 (1487)	**82.33% (80.73%)**
Case IV - 3Qs-MSG	31428 (7862)	572 (138)	13902 (3469)	18098 (4531)	70.83% (70.82%)
Case IV - 4Qs-MSG	29157 (7273)	2843 (727)	20799 (5161)	11201 (2839)	78.06% (77.71%)
Case IV - 6Qs-MSG	23377 (5722)	8623 (2278)	24081 (6007)	7919 (1993)	74.15% (73.31%)
Case IV - 7Qs-MSG	18991 (4627)	13009 (3373)	28579 (7100)	3421 (900)	74.33% (73.29%)

APPENDIX
ADDITIONAL QUALITATIVE ED RESULTS

Fig. 12 presents additional qualitative denoising results on other unseen published datasets of our proposed method compared to the state-of-the-art denoising models [20], [21].

REFERENCES

[1] G. Gallego *et al.*, "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, Jan. 2022.

[2] A. Glover and C. Bartolozzi, "Event-driven ball detection and gaze fixation in clutter," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 2203–2208.

[3] A. Amir *et al.*, "A low power, fully event-based gesture recognition system," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7388–7397.

[4] D. Bauer *et al.*, "Embedded vehicle speed estimation system using an asynchronous temporal contrast vision sensor," *J. Embedded Syst.*, vol. 2007, p. 082174, 2007, doi: 10.1155/2007/82174.

[5] H. Rebecq, G. Gallego, E. Mueggler, and D. Scaramuzza, "EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time," *Int. J. Comput. Vis.*, vol. 126, no. 12, pp. 1394–1414, Dec. 2018. [Online]. Available: https://jes-eurasipjournals.springeropen.com/articles/10.1155/2007/82174#citeas, doi: 10.1007/s11263-017-1050-6.

[6] A. Zhu *et al.*, "Ev-FlowNet: Self-supervised optical flow estimation for event-based cameras," in *Proc. Robot., Sci. Syst.*, Pittsburgh, PA, USA, Jun. 2018, pp. 1–9.

[7] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 1964–1980, Jun. 2019.

[8] A. Mitrokhin, C. Ye, C. Fermuller, Y. Aloimonos, and T. Delbruck, "EV-IMO: Motion segmentation dataset and learning pipeline for event cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 6105–6112.

[9] X. Huang *et al.*, "Real-time grasping strategies using event camera," *J. Intell. Manuf.*, vol. 33, no. 2, pp. 593–615, Feb. 2021.

[10] R. Muthusamy *et al.*, "Neuromorphic eye-in-hand visual servoing," *IEEE Access*, vol. 9, pp. 55853–55870, 2021.

[11] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 994–1001, Apr. 2018, doi: 10.1109/LRA.2018.2793357.

[12] C. Scheerlinck *et al.*, "CED: Color event camera dataset," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPR)*, 2019, pp. 1–10.

[13] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 494–514, Feb. 2021.

[14] A.-A. Liu, H. Tian, N. Xu, W. Nie, Y. Zhang, and M. Kankanhalli, "Toward region-aware attention learning for scene graph generation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 21, 2021, doi: 10.1109/TNNLS.2021.3086066.

[15] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Mar. 2021.

[16] S. Khan, M. Naseer, M. Hayat, S. Waqas Zamir, F. Shahbaz Khan, and M. Shah, "Transformers in vision: A survey," 2021, *arXiv:2101.01169*.

[17] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[18] K. Subramanyam Kalyan, A. Rajasekharan, and S. Sangeetha, "AMMUS : A survey of transformer-based pretrained models in natural language processing," 2021, *arXiv:2108.05542*.

[19] F.-J. Chang, M. Radfar, A. Mouchtaris, B. King, and S. Kunzmann, "End-to-end multi-channel transformer for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 5884–5888.

[20] R. W. Baldwin, M. Almatrafi, V. Asari, and K. Hirakawa, "Event probability mask (EPM) and event denoising convolutional neural network (EDnCNN) for neuromorphic cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1701–1710.

[21] Y. Feng, H. Lv, H. Liu, Y. Zhang, Y. Xiao, and C. Han, "Event density based denoising method for dynamic vision sensor," *Appl. Sci.*, vol. 10, no. 6, p. 2024, Mar. 2020.

[22] A. Khodamoradi and R. Kastner, "$O(N)$-space spatiotemporal filter for reducing noise in neuromorphic vision sensors," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 1, pp. 1–8, Jan./Mar. 2017.

[23] H. Liu, C. Brandli, C. Li, S.-C. Liu, and T. Delbruck, "Design of a spatiotemporal correlation filter for event-based sensors," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 722–725.

[24] V. Padala, A. Basu, and G. Orchard, "A noise filtering algorithm for event-based asynchronous change detection image sensors on TrueNorth and its implementation on TrueNorth," *Frontiers Neurosci.*, vol. 12, pp. 1–14, Mar. 2018.

[25] Y. Wang *et al.*, "EV-Gait: Event-based robust gait recognition using dynamic vision sensors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6351–6360.

[26] P. Duan, Z. W. Wang, X. Zhou, Y. Ma, and B. Shi, "EventZoom: Learning to denoise and super resolve neuromorphic events," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12819–12828.

[27] J. Chen, Y. Wang, Y. Cao, F. Wu, and Z.-J. Zha, "ProgressiveMotionSeg: Mutually reinforced framework for event-based motion segmentation," 2022, *arXiv:2203.11732*.

[28] T. Delbruck, "Frame-free dynamic digital vision," in *Proc. Intl. Symp. Secure-Life Electron., Adv. Electron. Quality Life Soc.*, 2008, pp. 21–26.

[29] S. Guo, Z. Kang, L. Wang, S. Li, and W. Xu, "HashHeat: An O(C) complexity hashing-based filter for dynamic vision sensor," in *Proc. 25th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2020, pp. 452–457.

[30] S. Guo *et al.*, "A noise filter for dynamic vision sensors using self-adjusting threshold," 2020, *arXiv:2004.04079*.

[31] R. Wes Baldwin, R. Liu, M. Almatrafi, V. Asari, and K. Hirakawa, "Time-ordered recent event (TORE) volumes for event cameras," 2021, *arXiv:2103.06108*.

[32] M. Kampffmeyer, Y. Chen, X. Liang, H. Wang, Y. Zhang, and E. P. Xing, "Rethinking knowledge graph propagation for zero-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11487–11496.

[33] K. Xu, H. Huang, P. Deng, and Y. Li, "Deep feature aggregation framework driven by graph convolutional network for scene classification in remote sensing," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 15, 2021, doi: 10.1109/TNNLS.2021.3071369.

[34] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[35] Z. Chen *et al.*, "Bridging the gap between spatial and spectral domains: A survey on graph neural networks," 2020, *arXiv:2002.11867*.

[36] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019, doi: 10.1145/3326362.

[37] R. Azzam, F. H. Kong, T. Taha, and Y. Zweiri, "Pose-graph neural network classifier for global optimality prediction in 2D SLAM," *IEEE Access*, vol. 9, pp. 80466–80477, 2021.
[38] M. Mansour, "A message-passing algorithm for graph isomorphism," 2017, *arXiv:1704.00395*.
[39] N. Carion *et al.*, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.
[40] M. C. H. Lee, K. Petersen, N. Pawlowski, B. Glocker, and M. Schaap, "TeTrIS: Template transformer networks for image segmentation with shape priors," *IEEE Trans. Med. Imag.*, vol. 38, no. 11, pp. 2596–2606, Nov. 2019.
[41] Accessed: Aug. 20, 2021. [Online]. Available: https://www.universal-robots.com/products/ur10-robot/
[42] (Mar. 2021). *Inivation Launches Next-Generation Event-Based Dynamic Vision Sensor*. [Online]. Available: https://inivation.com/inivation-launches-next-generation-event-based-dynamic-vision-sensor/
[43] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI–8, no. 6, pp. 679–698, Nov. 1986.
[44] P. Bergström and O. Edlund, "Robust registration of point sets using iteratively reweighted least squares," *Comput. Optim. Appl.*, vol. 58, no. 3, pp. 543–561, Jul. 2014.
[45] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.
[46] J. Lei Ba, J. Ryan Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
[47] *Pytorch*. Accessed: Oct. 10, 2021. [Online]. Available: https://pytorch.org/
[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[49] A. Vitale, A. Renner, C. Nauer, D. Scaramuzza, and Y. Sandamirskaya, "Event-driven vision and control for UAVs on a neuromorphic chip," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 103–109.
[50] N. J. Sanket *et al.*, "EVDodgeNet: Deep dynamic obstacle dodging with event cameras," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 10651–10657.
[51] M. Salah *et al.*, "A neuromorphic vision-based measurement for robust relative localization in future space exploration missions," 2022, *arXiv:2206.11541*.
[52] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll, "Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 34–49, Jul. 2020.
[53] G. Chen *et al.*, "Neuromorphic vision-based fall localization in event streams with temporal-spatial attention weighted network," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 1–12, Sep. 2022.

Yusra Alkendi received the M.Sc. degree in mechanical engineering from the Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates, in 2019, where she is currently pursuing the Ph.D. degree in aerospace engineering with the Khalifa University Center for Autonomous Robotics Systems (KUCARS) with a focus on robotics.

Her current research is focused on the application of artificial intelligence (AI) in the fields of dynamic vision for perception and navigation.

Rana Azzam received the B.Sc. degree in computer engineering, the M.Sc. degree in electrical and computer engineering, and the Ph.D. degree in engineering with a focus on robotics from Khalifa University, Abu Dhabi, United Arab Emirates, in 2014, 2016, and 2020, respectively.

She is currently a Post-Doctoral Fellow with the Department of Aerospace Engineering, Khalifa University. Her research interests include machine learning, reinforcement learning, navigation, and simultaneous localization and mapping.

Abdulla Ayyad (Member, IEEE) received the M.Sc. degree in electrical engineering from The University of Tokyo, Tokyo, Japan, in 2019.

He conducted research with the Spacecraft Control and Robotics Laboratory, The University of Tokya. He is currently a Research Associate with the Khalifa University Center for Autonomous Robotic Systems (KUCARS) and the Aerospace Research and Innovation Center (ARIC), Abu Dhabi, United Arab Emirates, working on several robot autonomy projects. His current research interest includes the application of artificial intelligence (AI) in the fields of perception, navigation, and control.

Sajid Javed received the B.Sc. degree in computer science from the University of Hertfordshire, Hatfield, U.K, in 2010, and the combined master's and Ph.D. degree in computer science from Kyungpook National University, Daegu, Republic of Korea, in 2017.

He is currently an Assistant Professor of computer vision with the Department of Electrical Engineering and Computer Science (EECS), Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates. Prior to that, he was a Research Scientist at the Khalifa University Center for Autonomous Robotics System (KUCARS) from 2019 to 2021. Before joining Khalifa University, he was a Research Fellow with the University of Warwick, Coventry, U.K, from 2017 to 2018, where he worked on histopathological landscapes for better cancer grading and prognostication. His research interests include visual object tracking in the wild, multiobject tracking, background–foreground modeling from video sequences, moving object detection from complex scenes, and cancer image analytics, including tissue phenotyping, nucleus detection, and nucleus classification problems. His research themes involve developing deep neural networks, subspace learning models, graph neural networks, and vision transformers.

Dr. Javed is also the Area Chair of the Asian Conference on Computer Vision (ACCV).

Lakmal Seneviratne is currently a Professor of mechanical engineering and the Founding Director of the Centre for Autonomous Robotic Systems (KUCARS), Khalifa University, Abu Dhabi, United Arab Emirates. He is also an Emeritus Professor at King's College London, London, U.K. He has worked as an Associate Provost for Research and Graduate Studies and an Associate VP Research at Khalifa University. Prior to joining Khalifa University, he was a Professor of mechatronics, the Founding Director of the Centre for Robotics Research, and the Head of the Division of Engineering, King's College London. His main research interests include robotics and automation, with particular emphasis on increasing the autonomy of robotic systems interacting with complex dynamic environments. He has published over 400 peer-reviewed publications on these topics.

Dr. Seneviratne is a member of the Mohammed Bin Rashid Academy of Scientists in United Arab Emirate.

Yahya Zweiri (Member, IEEE) received the Ph.D. degree from King's College London, London, U.K., in 2003.

He is currently an Associate Professor with the Department of Aerospace Engineering and the Deputy Director of Advanced Research and Innovation Center, Khalifa University, Abu Dhabi, United Arab Emirates. He was involved in defense and security research projects in the last 20 years at the Defense Science and Technology Laboratory, King's College London, and the King Abdullah II Design and Development Bureau, Amman, Jordan. He has published over 110 refereed journals and conference papers and filed ten patents in USA and U.K., in the unmanned systems field. His research interests include robotic systems for extreme conditions with particular emphasis on applied artificial intelligence (AI) aspects and neuromorphic vision systems.