Hypergraph-Based Multi-View Action Recognition Using Event Cameras

Yue Gao, *Senior Member, IEEE*, Jiaxuan Lu, Siqi Li, Yipeng Li, and Shaoyi Du, *Member, IEEE*

***Abstract*—Action recognition from video data forms a cornerstone with wide-ranging applications. Single-view action recognition faces limitations due to its reliance on a single viewpoint. In contrast, multi-view approaches capture complementary information from various viewpoints for improved accuracy. Recently, event cameras have emerged as innovative bio-inspired sensors, leading to advancements in event-based action recognition. However, existing works predominantly focus on single-view scenarios, leaving a gap in multi-view event data exploitation, particularly in challenges like information deficit and semantic misalignment. To bridge this gap, we introduce *HyperMV*, a multi-view event-based action recognition framework. HyperMV converts discrete event data into frame-like representations and extracts view-related features using a shared convolutional network. By treating segments as vertices and constructing hyperedges using rule-based and KNN-based strategies, a multi-view hypergraph neural network that captures relationships across viewpoint and temporal features is established. The vertex attention hypergraph propagation is also introduced for enhanced feature fusion. To prompt research in this area, we present the largest multi-view event-based action dataset $\mathbf{THU}^{\mathbf{MV-EACT}}\mathbf{-50}$, comprising 50 actions from 6 viewpoints, which surpasses existing datasets by over tenfold. Experimental results show that HyperMV significantly outperforms baselines in both cross-subject and cross-view scenarios, and also exceeds the state-of-the-arts in frame-based multi-view action recognition.**

***Index Terms*—Multi-view action recognition, event camera, dynamic vision sensor, hypergraph neural network.**

I. Introduction

ACTION recognition, a fundamental task in computer vision, involves automatically identifying and classifying human actions from video data. It has gained significant attention due to its broad applications in various fields, including surveillance, human-computer interaction, robotics, and video content analysis [1], [2], [3], [4]. Accurate action recognition enables intelligent systems to understand human behavior, facilitate human-centric applications, and enhance the interaction between humans and machines [5]. With the rapid advancement of multi-camera systems and virtual reality technologies, there is an increasing need to capture and analyze actions from multiple viewpoints. Single-view action recognition methods [1], [6], [7], [8] are inherently limited by the viewpoint from which the action is observed. This viewpoint dependency often leads to incomplete understanding and potential misclassification of actions. In contrast, multi-view action recognition approaches [9], [10] offer distinct advantages by integrating information from different viewpoints, which can capture complementary information, leading to more accurate recognition results.

Both single-view and multi-view action recognition methods predominantly rely on traditional frame-based cameras and data. As an alternative, bio-inspired event cameras, e.g., Dynamic Vision Sensors (DVS) [11], [12] have emerged as promising vision sensors in recent years. Unlike traditional frame-based cameras that capture images at a fixed exposure rate, event cameras operate by asynchronously detecting changes in brightness at each pixel. This asynchronous event-based feature offers several advantages, including high temporal resolution, low power consumption, and the potential for privacy encryption. While several methods have explored action recognition using event data in single-view settings [13], [14], [15], [16], there is currently no existing work that utilizes multi-view event data for action recognition to the best of our knowledge.

One of the reasons is the lack of datasets. Although there are existing single-view event action datasets [16], [17], [18], [19], there is a lack of comprehensive multi-view event datasets specifically designed for action recognition. DHP19 [18] is the only dataset that can be used for multi-view event-based action recognition, but it is oriented towards pose estimation tasks and is small in scale (33 actions and 2,228 recordings). To facilitate research in multi-view event-based action recognition, we introduce the $\mathbf{THU}^{\mathbf{MV-EACT}}\mathbf{-50}$, an expansion of the single-view $\text{THU}^{\text{E-ACT}}$-50 [16], by incorporating more viewpoints. The $\text{THU}^{\text{MV-EACT}}$-50 dataset comprises 50 distinct actions observed from 6 different views, encompassing 4 frontal and 2 backal views, resulting in a comprehensive collection of 31,500 recording sequences. The captured dataset stands as the largest multi-view event-based action dataset available to date, which will be released after acceptance.

Manuscript received 28 December 2023; revised 14 March 2024; accepted 21 March 2024. Date of publication 27 March 2024; date of current version 5 September 2024. This work was supported in part by the National Science and Technology Major Project of China under Grant 2020AAA0108102, and in part by the National Natural Science Funds of China under Grant 62088102 and Grant 62021002. Recommended for acceptance by Cees G.M. Snoek. *(Corresponding authors: Jiaxuan Lu; Shaoyi Du.)*

Yue Gao and Siqi Li are with the BNRist, THUIBCS, BLBCI, School of Software, Tsinghua University, Beijing 100084, China (e-mail: gaoyue@tsinghua.edu.cn; lsq19@mails.tsinghua.edu.cn).

Jiaxuan Lu is with Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China (e-mail: lujiaxuan@pjlab.org.cn).

Yipeng Li is with the BNRist, THUIBCS, BLBCI, Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: liep@tsinghua.edu.cn).

Shaoyi Du is with the Department of Ultrasound, The Second Affiliated Hospital of Xi'an Jiaotong University, Xi'an 710006, China, and also with the National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, National Engineering Research Center for Visual Information and Applications, and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: dushaoyi@xjtu.edu.cn).

Digital Object Identifier 10.1109/TPAMI.2024.3382117

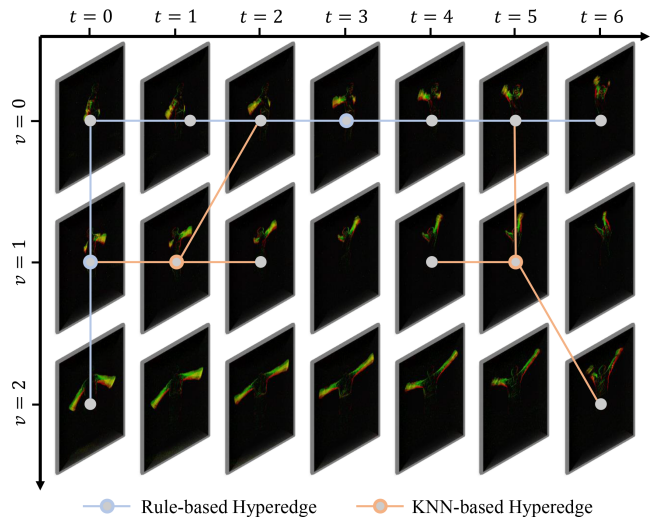

Fig. 1. Addressing information deficit and semantic misalignment in multi-view event-based action recognition, the proposed multi-view hypergraph neural network leverages rule-based and KNN-based hyperedges to correlate features across views and temporal segments.

In terms of the method, multi-view event data has a serious information deficit compared to frame-based images, since event data only records the regions of motion rather than invariant background. Moreover, event data captured from different viewpoints often encounters a challenge known as semantic misalignment (shown in Fig. 1). Semantic misalignment refers to the inconsistency in pixel position when representing the same region of a person from various viewpoints, a prevalent issue in multi-view scenarios. Therefore, effective fusion of features from diverse viewpoints and moments is pivotal in enhancing the accuracy of multi-view action recognition. To address these challenges, we propose a hypergraph-based framework called *HyperMV* for multi-view event-based action recognition, as shown in Fig. 2. The proposed approach explores the high-order associations between viewpoint and temporal features and leverages these associations to facilitate feature fusion. Specifically, discrete event data are first processed into frame-like intermediate representations that are fed into the view feature extraction module. A shared convolutional network is acted upon for each view to extract view-related features. In the subsequent stage, each temporal segment under each view is considered as a vertex. By employing both rule-based and K-nearest neighbors (KNN) strategies to construct hyperedges, we establish a multi-view hypergraph neural network to capture both explicit and implicit relationships among viewpoint and temporal features. Vertex attention hypergraph propagation is also proposed for better feature fusion. In the final stage, each vertex is assigned a weight to generate the final embedding, which is subsequently used for action classification. Extensive experiments involving both cross-subject and cross-view scenarios demonstrate significant improvements compared to the baseline approaches.

Overall, the main contributions of this paper can be summarized as follows:

- We extend single-view THU$^{\text{E-ACT}}$-50 [16] to contribute the $\mathbf{THU^{MV-EACT-50}}$ dataset, the largest multi-view event action dataset to date, comprising 50 actions from 6 viewpoints, providing a valuable resource for evaluating algorithms in multi-view event-based action recognition. The constructed benchmarks can be accessed at: https://gaoyue.org/dataset/THU-MV-EACT-50.
- We propose a framework called *HyperMV* using the multi-view hypergraph neural network for event-based action recognition, effectively fusing features from different viewpoints and temporal segments.
- Through experiments in both cross-subject and cross-view scenarios, we demonstrate the effectiveness of our method with significant improvements in multi-view event-based action recognition compared to the baseline approaches.

II. Related Work

A. Frame-Based Action Recognition

Significant strides have been made in the field of frame-based action recognition [20], [21], [22]. Tran et al. introduced C3D [23], a 3D CNN model that merges appearance features with motion data for video sequences. Sun et al. [24] employed factorization techniques to break down 3D convolution kernels and utilized spatio-temporal features across different CNN layers. The concept of the two-stream CNN [25] was first introduced to extract features from keyframes and the optical flow channel. Wang et al. further developed the Temporal Segment Network (TSN) [26] to utilize video segments within the two-stream CNN framework. In terms of the multi-branch structure, Feichtenhofer et al. [27] proposed a single CNN that merges spatial and temporal features before the final layers, yielding impressive results. Wang et al. [28] introduced a multi-branch neural network where each branch handles different levels of features. For the fusion of features at different sample rates, Feichtenhofer et al. proposed the SlowFast [29] which achieves performance improvements by setting fast and slow pathways. In brief, the progression of single-view action recognition is largely dependent on the enhanced aggregation of spatial-temporal features. However, these studies were based on information from a single viewpoint and thus did not learn features from multiple viewpoints.

When it comes to multi-view action recognition, where videos come from various viewpoints, earlier action recognition methods that only utilize view-invariant representations may not deliver optimal results [30], [31]. Liu et al. [32] introduced a genetic algorithm that merges features from various views through a process of iterative evolution. Drawing inspiration from subspace learning, Kong et al. [33] developed a projection matrix to map features from different views into a shared subspace. In an effort to further progress multi-view learning, Nie et al. [34] endeavored to autonomously learn the optimal weight of each viewpoint without the need for additional parameters. Ullah et al. [35] present a conflux Long Short-Term Memory (LSTM) network to recognize actions from multi-view cameras. For improvement, Bai et al. [36] put forth a collaborative attention mechanism to discern the attention disparities among multi-view inputs. Shah et al. [37] employ supervised contrast learning to learn feature embedding robust to changes in viewpoint. Another category of methodologies [38], [39] made use of Generative Adversarial

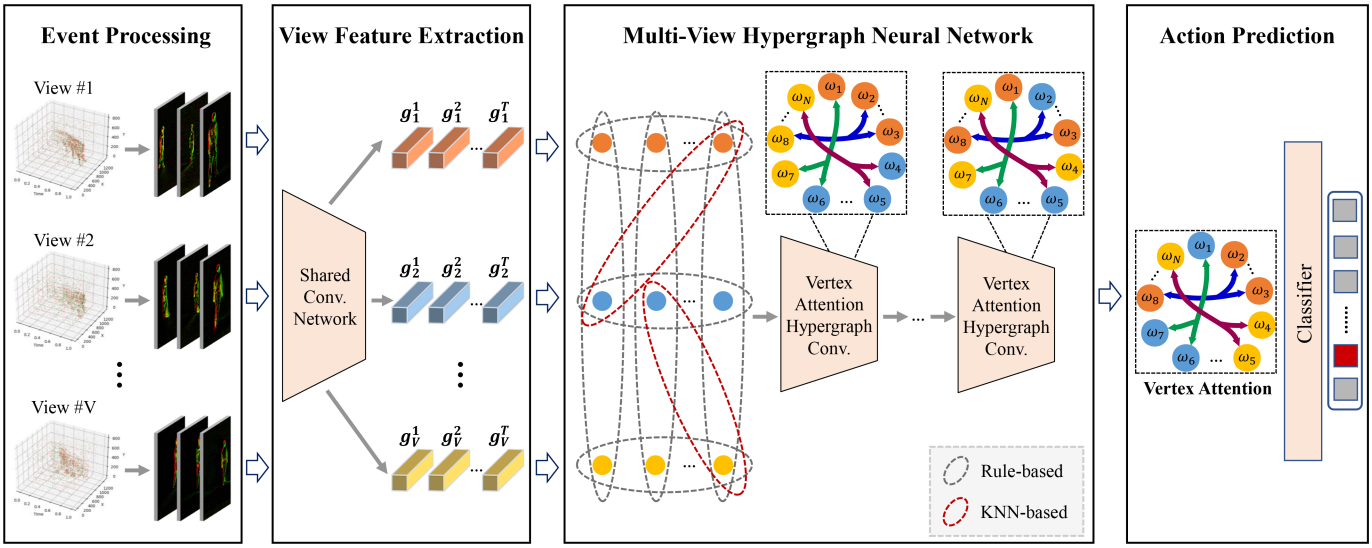

Fig. 2. Pipeline of the proposed multi-view event-based action recognition framework, including Event Processing, View Feature Extraction, Multi-View Hypergraph Neural Network, and Action Prediction.

Networks (GANs) to generate one view conditional on another, thereby probing the potential interconnections between views. Hence, the key to effective multi-view action recognition lies in the fusion of diverse information across various views.

B. Event-Based Action Recognition

Event cameras are bio-inspired vision sensors that asynchronously detect brightness changes at each pixel. For a given pixel located at (x, y), an event is triggered at a specific timestamp t when the following condition is met:

$$L(x, y, t) - L(x, y, t') > p \cdot \theta, \tag{1}$$

where $L(x, y, t)$ represents the logarithm of the brightness, while t' is the timestamp of the last event that was triggered at the (x, y) location. The constant θ serves as a threshold, and $p \in \{-1, +1\}$ denotes the polarity of the event, indicating whether the brightness is increasing or decreasing. Events are recorded as tuples $e = (x_k, y_k, t_k, p_k)$, where x_k and y_k are pixel coordinates, t_k is the timestamp, and p_k is the polarity. Compared to traditional frame-based imaging devices, event cameras do not record color or texture, providing a unique privacy advantage. Moreover, they only activate upon significant intensity changes, leading to reduced power usage, about 150 times lower than regular cameras [40]. Additionally, in scenarios involving high-speed motion, event cameras can capture motion without the usual blur seen in frame-based cameras.

However, since event cameras are new vision sensors introduced in recent years, only a few works have explored event-based action recognition. Xiao et al. introduced the HMAX Spiking Neural Network (HMAX SNN) [13] to extract temporal features via multispike encoding. Building upon this, Liu et al. put forth Motion SNN [14], which leverages motion information to construct a multilayer SNN structure. Chen et al. [15] proposed to view events as 3D points and input them into a dynamic graph CNN for gesture recognition. To leverage the powerful learning capabilities of CNNs in image-related tasks, several studies have transformed the discrete event data into frame-like representations. Ghosh et al. [41] introduced spatio-temporal filtering in the spike-event domain, with the resulting representations inputted into CNNs. Innocenti et al. [42] proposed the conversion of event data into Temporal Binary Representation for subsequent action recognition using CNNs. Wu et al. [43] accumulated the event data into frames and leveraged the multi-path deep neural network for action recognition. Gao et al. [16] proposed to fuse multiple event representations in a learnable manner and feed them into the event-based slow-fast network for action recognition. Nonetheless, these methods are all tailored to single-view event-based action recognition, and there is no work exploring multi-view action recognition based on event data to the best of our knowledge.

C. Graph and Hypergraph Neural Network

In recent years, Graph Neural Networks (GNNs) [44], [45] and their variants [46], [47], [48], [49] have emerged as powerful tools in the realm of data analysis, demonstrating their versatility in a broad spectrum of graph-structured tasks, including graph classification [50], [51], [52], graph clustering [53], [54], [55], and graph link prediction [56], [57]. The power of GNNs also extends beyond graph-structured data, as they have also been effectively utilized in non-graph structured data. This includes areas such as document classification [58], image classification [59], [60], person re-identification [61], [62], and action recognition [63], [64]. Since the GNNs are inherently limited by the graph structure that only allows for one-to-one relationships between vertices, some researchers have turned to hypergraphs [65] and Hypergraph Neural Networks (HGNNs) [66]. These advanced structures and networks extend the concept of graphs by allowing hyperedges to connect multiple vertices, capable of constructing and learning high-order complex relationships among vertices. In the context of multi-view

event-based action recognition tasks, the features on both the viewpoint and temporal dimensions often suffer from severe information deficit and semantic misalignment. Given these challenges, we posit that GNNs and HGNNs could potentially demonstrate their robust capabilities in association modeling.

D. Datasets for Action Recognition

Datasets serve as crucial catalysts in advancing deep learning methodologies. In the realm of frame-based action recognition, numerous well-established datasets already exist. The KTH dataset [67], an early action dataset, comprises videos of 6 action categories at a resolution of 160×120 across various scenes. The I3DPost dataset [68] offers videos of two individuals interacting, performing 8 different actions. The UCF50 and UCF101 datasets [69] encompass 50 and 101 action categories, respectively, sourced from YouTube. The Kinetics dataset [1], a series of large-scale datasets released by DeepMind, contains 400 action categories with over 400 videos per action. As for the multi-view action recognition, the NUCLA dataset [70] is captured in UCLA from three different viewpoints, covering 10 action categories performed by 10 subjects. The NTU dataset [71] stands out with its integration of RGB, depth, and infrared sensors to capture 60 action classes from multiple angles, consisting of 56,880 videos. The PKU-MMD dataset [72] offers a large-scale benchmark for continuous action recognition, containing 1,076 long video sequences in 51 action categories in 3 camera views. The UESTC dataset [73] consists of 25,000 sequences across 40 action categories with 8 static viewpoints. The ETRI dataset [74] is a multi-view action recognition dataset for elderly care, which has 112,620 videos captured from 55 action classes across 8 viewpoints.

Despite the abundance of conventional frame-like datasets, there is a noticeable scarcity of event-based action recognition datasets. As for the simulated datasets, N-EPIC-Kitchens [75] is an event version of the EPIC-Kitchens generated by the event camera simulator. The event UCF-50 [76] is derived from the UCF-50 action recognition dataset, which was captured by displaying its data on a monitor. Regarding the real-world event-based action recognition dataset, PAF [17] is the first one, which offers 450 recordings spanning 10 categories from an indoor office setting, each with an average length of $5s$ and a spatial resolution of 346×260. N-HAR [19] is another indoor dataset with 3,091 videos, but it is category-unbalanced and contains only 5 actions. DailyAction [14] provides 1,440 recordings across 12 action categories, albeit with a limited spatial resolution of 128×128 due to acquisition via DVS128 [77]. THU$^{\text{E-ACT}}$-50 [16] vastly expands the scale of data used for single-view action recognition to include 50 action categories and a total of 10,500 recordings. DHP19 [18] is currently the only dataset available for multi-view event-based action recognition, including 33 sub-actions and a total of 2,228 recordings. However, DHP19 is primarily designed for the pose estimation task, and the actions are all localized limb movements (e.g., left arm abduction, right arm abduction, left leg knee lift, right leg knee lift) rather than human actions applicable to everyday scenes. Therefore, there exists a pressing demand for large-scale multi-view action recognition datasets captured by event cameras.

III. Method

To confront the challenges of information deficit and semantic misalignment, we introduce a hypergraph-based framework for multi-view event-based action recognition, as depicted in Fig. 2. Initially, the event processing module transforms discrete event data into frame-like intermediate representations. Afterward, the view feature extraction module extracts view-related features through a shared convolutional network for each viewpoint. Each temporal segment under each view is considered as a vertex, and the multi-view hypergraph neural network based on rule-based and KNN-based strategies is employed to capture both explicit and implicit relationships. The vertex attention mechanism is also utilized in both the proposed vertex attention hypergraph propagation and the final vertex weighting operator, thereby generating the ultimate embedding for action recognition.

A. Event Processing

There are two primary strategies in the event data processing. One utilizes Spiking Neural Networks (SNNs) to process event data as impulses [13], [78], [79], [80]. However, SNNs have limited learning abilities. Alternatively, some methods transform event data into intermediate representations [81], [82], [83], [84], thereby harnessing the advanced learning capabilities of Convolutional Neural Networks (CNNs). In our approach, we follow the latter method and transform the raw event data into the widely used Event Frame [85]. For a given view v, the stream of events E_v is decomposed into a sequence of T event packets in temporal order, denoted as $E_v = \{E_v^t\}_{t=1}^T$. Each event packet E_v^t represents the set of events collected within the time interval from $t-1$ to t, represented as

$$E_v^t = \{(x_k, y_k, t_k, p_k)\}_{k=1}^N, \tag{2}$$

where N is the total number of events within the time interval from $t-1$ to t.

Subsequently, we generate the event frame I_v^t from the event packet E_v^t by summing the events triggered at each pixel location for the two polarities, denoted as

$$I_v^t(x, y) = \sum_{e \in E_v^t} p_k \cdot \delta(x - x_k, y - y_k), \tag{3}$$

where $\delta(\cdot)$ denotes the Dirac delta function, which equals 1 when $x = x_k$ and $y = y_k$, and 0 otherwise. As a result, for each view, the raw event data is transformed into a frame-like intermediate representation $I_v = \{I_v^1, I_v^2, \ldots, I_v^T\}$ with dimensions (X, Y, T). The event frame is straightforward yet effective, as it encapsulates both spatial and temporal information, which is crucial for action recognition.

B. View Feature Extraction

As for the view feature extraction module, our objective is to obtain a comprehensive set of features from different

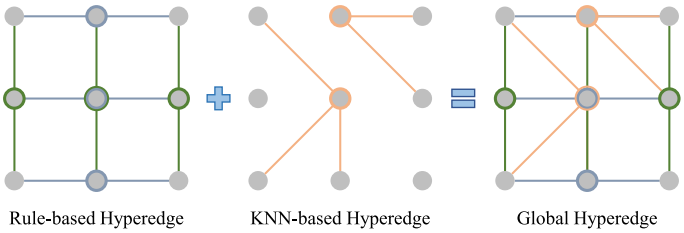

Fig. 3. Rule-based and KNN-based hyperedges are combined to model explicit and implicit associations of features.

viewpoints. The module operates on input intermediate representations of events, denoted as $I = (I_1, I_2, \ldots, I_V)$. Each I_v corresponds to the intermediate representation from viewpoint v. Afterwards, I is processed by a shared convolutional network, which consists of a series of convolutional layers acting as a backbone and a global pooling layer. The shared convolutional backbone is designed to reduce the spatial resolution of each viewpoint representation, thus effectively concentrating vital view-related information. The output can be represented as $C = (C_1, C_2, \ldots, C_V)$, where $C_v = \{C_v^t\}_{t=1}^T$ contains the feature maps from viewpoint v. For each view v and moment t, the global pooling layer is then applied to the convolved feature map c_v^t to obtain a one-dimensional embedding, denoted as g_v^t.

To formalize, for a given viewpoint v and a moment t, the one-dimensional embedding is given by

$$g_v^t = \text{Pool}\left(\text{Convs}\left(I_v^t\right)\right), \tag{4}$$

where $\text{Convs}(\cdot)$ represents the shared convolutional backbone applied to the intermediate representation I_v^t, and $\text{Pool}(\cdot)$ represents the global pooling layer. The features g_v^t encapsulate the view-related features of each viewpoint for the moment t, which provides a representative embedding for further aggregation at subsequent stages.

C. Multi-View Hypergraph Construction

Considering the challenges posed by information deficit and semantic misalignment inherent in multi-view event-based action recognition, the strategy employed for fusing features from various viewpoints and across different temporal segments can greatly influence performance. In a multi-view scenario, there exist both sequential associations across different moments within a single view, and correlations between different views at the same instant. Compared with a normal graph that can only model one-to-one associations, a hypergraph extends the structure of the graph so that multiple vertices can be connected using a single hyperedge. Accordingly, we propose the multi-view hypergraph neural network to integrate features across viewpoints and temporal segments, i.e., using rule-based hyperedges to establish explicit connections, and using KNN-based hyperedges to model implicit connections, as illustrated in Fig. 3.

Specifically, we regard the one-dimensional features g_v^t under the view v and moment t as the vertices, denoted as (v, t). In the multi-view scenario with V viewpoints and T time windows, there are a total of $V \times T$ vertices. For the rule-based strategy, we employ two types of hyperedges: the time-consistent hyperedge $\mathcal{E}_{rule}^{(t)}$ connects vertices of different moments of the same view, and the view-consistent hyperedge $\mathcal{E}_{rule}^{(v)}$ links vertices from varying views at an identical moment, denotes as

$$\mathcal{E}_{rule}^{(t)} = \{(v,t), \forall(v',t') \mid v = v', t \neq t'\}, \tag{5}$$

$$\mathcal{E}_{rule}^{(v)} = \{(v,t), \forall(v',t') \mid t = t', v \neq v'\}. \tag{6}$$

Then, the rule-based hyperedges can be denoted as $\mathcal{E}_{rule} = \mathcal{E}_{rule}^{(t)} \cup \mathcal{E}_{rule}^{(v)}$. In terms of the KNN-based strategy, we identify for each vertex (v, t) the k vertices in the embedding that exhibit the highest similarity, without consideration for perspective or temporal ordering. These identified vertices are then connected using a hyperedge. As such, the KNN-based hyperedge set $\mathcal{E}_{knn}$ is denoted as

$$\mathcal{E}_{knn} = \{(v,t), \forall(v',t) \in N_k(v,t)\}, \tag{7}$$

where $N_k(v, t)$ signifies the k vertices demonstrating the highest similarity to vertex (v, t) in terms of their embeddings. Subsequently, the two types of hyperedge sets are combined to obtain the global hyperedge set $\mathcal{E} = \mathcal{E}_{rule} \cup \mathcal{E}_{knn}$. Unlike graphs, hypergraphs utilize the incidence matrix $\mathbf{H}$ to indicate whether the hyperedges $e \in \mathcal{E}$ contain the vertices (v, t), which can be expressed as

$$\mathbf{H}((v,t),e) = \begin{cases} 1, & (v,t) \in e \\ 0, & (v,t) \notin e \end{cases}. \tag{8}$$

D. Vertex Attention Hypergraph Propagation

After the multi-view hypergraph is constructed, the features of the vertices are updated iteratively based on the connectivity of the hyperedges. While the foundational work on Hypergraph Neural Networks (HGNNs) [66] provides a formula for feature propagation through the hypergraph convolutional layer, it solely accounts for the weights associated with the hyperedges, disregarding the weights assigned to the vertices. In our perspective, vertices across distinct viewpoints and moments should possess diverse amounts of information, particularly in the context of event data. With this goal in mind, we propose the vertex attention hypergraph propagation based on the original one, which can be mathematically expressed as

$$\mathbf{X}^{(l+1)} = \sigma\left(\mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}\mathbf{W}_e\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{W}_v\mathbf{D}_v^{-\frac{1}{2}}\mathbf{X}^{(l)}\Theta^{(l)}\right), \tag{9}$$

where $\mathbf{X}^{(l)}$ corresponds to the vertex features of the l^{th} layer. The diagonal matrices, $\mathbf{D}_v$ and $\mathbf{D}_e$, are formulated from the degree of the vertex and the degree of the hyperedges respectively, and are computed based on the correlation matrix $\mathbf{H}$. The nonlinear activation function is represented as $\sigma(\cdot)$. The trainable parameters include $\mathbf{W}_e$, $\mathbf{W}_v$, and $\Theta^{(l)}$. Here, $\mathbf{W}_e$ functions as the weight matrix corresponding to each hyperedge, $\mathbf{W}_v$ represents the weight matrix associated with each vertex, while $\Theta^{(l)}$ denotes the weight matrix utilized for feature extraction at the l^{th} layer.

To delve deeper into the feature propagation mechanisms inherent in the vertex attention hypergraph convolution layer, we refer to Fig. 4 which illustrates the vertex-hyperedge-vertex feature fusion process. Initially, the vertex features $\mathbf{X}^{(0)}$ are passed through the first fully connected layer with weight $\Theta^{(0)}$ applied

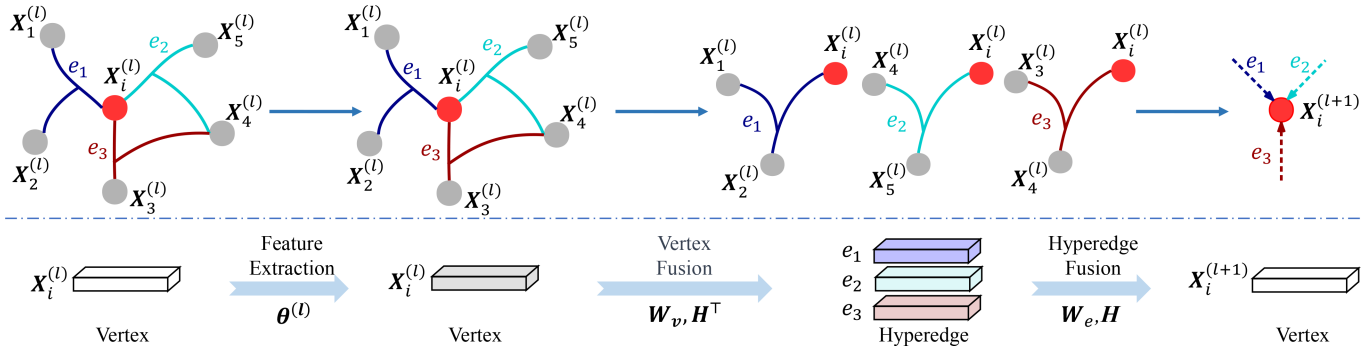

Fig. 4. Vertex attention hypergraph propagation via the vertex-hyperedge-vertex process: (1) Initial vertex features enter a fully connected layer for feature extraction. (2) Hyperedge features are then weighted from these vertex features. (3) Subsequently, these hyperedge features are fused together to formulate the vertex features for the next layer.

to extract relevant features. Subsequently, these extracted features are weighted by $\mathbf{W}_v$ to generate each hyperedge's features, represented as the product of the vertex feature matrix $\mathbf{X}^{(0)}$ and the transpose of the incidence matrix $\mathbf{H}^\top$. Subsequently, the hyperedge features are weighted by $\mathbf{W}_e$ to produce the updated vertex features for the next layer $\mathbf{X}^{(1)}$, encapsulated by the product of the hyperedge weights $\mathbf{W}_e$ and the incidence matrix $\mathbf{H}$. Throughout the propagation process, there exists a dynamic interplay between vertex and hyperedge features, which amplifies the capacity to capture intricate high-order relationships. Finally, upon completion of L rounds of hypergraph convolution, we obtain the final vertex features $\mathbf{X}^{(L)}$.

E. Action Prediction

In the pursuit of effective action classification, it is crucial to merge the final features of $V \times T$ vertices. To this end, we assign different weights to vertices for a superior graph-level representation. Specifically, assuming that the vertex features obtained after the hypergraph convolution layers are represented as $\mathbf{X}^{(L)} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, where $N = V \times T$, the vertex weighting operator computes the attention weight ω_i for each individual vertex. Consequently, the features of each vertex are weighted and amalgamated with their corresponding attention weight, leading to a graph-level feature representation $\mathbf{x}_g$, denoted as

$$\mathbf{x}_g = \sum_{i=1}^{N} \omega_i \mathbf{x}_i, \tag{10}$$

where

$$\omega_i = \frac{\|\mathbf{x}_i\|_1}{\sum_j^N \|\mathbf{x}_j\|_1}. \tag{11}$$

Finally, the resulting graph-level feature representations $\mathbf{x}_g$ are fed into the fully connected layer for the prediction of action categories. The cross-entropy loss function is used for training.

IV. THU$^{\mathrm{MV-EACT}}$-50 Benchmark

Regardless of single-view or multi-view scenes, the current event-based datasets are lacking in terms of both action categories and data scale, and contain too simplified actions to meet the data requirements for practical applications of action recognition systems. As shown in Table I, our previously released THU$^{\text{E-ACT}}$-50 [16] dataset expands the number of action categories to 50 and recordings to 10,500 for single-view event-based action recognition. However, with respect to multi-view scenarios, DHP19 [18] remains the only applicable choice at present. Although it includes 33 categories of body movements, it focuses on human pose estimation and the simplicity of its limb movement categories restricts its practical applicability.

Given these circumstances, we have extended the single-viewpoint THU$^{\text{E-ACT}}$-50 [16] dataset and are about to release the THU$^{\text{MV-EACT}}$-50 dataset, which is the first large-scale multi-view dataset specifically for the event-based action recognition task, and also the largest event action dataset to date. The THU$^{\text{MV-EACT}}$-50 comprises 50 action categories, 31,500 recordings, and 6 viewpoints at a resolution of 1280×800, which surpasses the scale of DHP19 [18] by factors of 14, as shown in Table I. The THU$^{\text{MV-EACT}}$-50 has the same 50 action categories as the THU$^{\text{E-ACT}}$-50, including actions for indoor health monitoring (e.g., falling down, headache, stomachache, back pain, vomit, staggering, etc.), whole-body movements (e.g., walking, running, jump up, running in circles, squat down, tai chi, etc.) and detail-sensitive actions (e.g., nod head, shake head, thumb up, clap, rub hands, wipe face, etc.). At the same time, some confusing action groups are added to increase the difficulty (e.g., stand up, sit down, sit and then stand, etc.). In addition to single-person actions, the dataset also includes actions for interactions between people and objects (e.g., calling with phone, swinging objects, throw, pick up, drink water, open/close umbrella, put on/take off glasses, put on/take off bag, etc.) and actions for interactions between two people (e.g., shake hands, fighting, handing objects, lifting chairs). The complete list of actions is shown in Table II. The props used in the acquisition process include books, umbrellas, school bags, fans, glasses, cups, and tissues.

For the acquisition environment, the THU$^{\text{MV-EACT}}$-50 dataset is collected using CeleX-V [86], through 6 event cameras with different viewpoints arranged across an indoor venue of approximately $100m^2$. The event cameras, each held by a tripod approximately $1m$ above the ground, afford 4 frontal and 2 backward views of the performer, as shown in Fig. 6. Each event camera is

TABLE I
COMPARISONS BETWEEN THE THU$^{\text{MV-EACT}}$-50 BENCHMARK AND OTHER EXISTING DATASETS FOR SINGLE-VIEW AND MULTI-VIEW EVENT-BASED ACTION RECOGNITION

Type	Benchmark	View Num.	Category Num.	Recording Num.	Subject Num.	Resolution	Sensor
Single-view	PAF [17]	-	10	450	15	346 × 260	DAVIS 346
	DailyAction [14]	-	12	1,440	15	128 × 128	DVS 128
	N-HAR [19]	-	5	3,091	30	304 × 240	ATIS
	THU$^{\text{E-ACT}}$-50 [16]	-	50	10,500	105	1280 × 800	CeleX-V
Multi-view	DHP19 [18]	4	33	2,228	17	346 × 260	DAVIS 346
	THU$^{\text{MV-EACT}}$-50	**6**	**50**	**31,500**	**105**	1280 × 800	CeleX-V

TABLE II
LIST OF ACTIONS IN THE CONSTRUCTED THU$^{\text{MV-EACT}}$-50 BENCHMARK

A0: Walking	A10: Cross arms	A20: Calling with phone	A30: Fan	A40: Check time
A1: Running	A11: Salute	A21: Reading	A31: Open umbrella	A41: Drink water
A2: Jump up	A12: Squat down	A22: Tai chi	A32: Close umbrella	A42: Wipe face
A3: Running in circles	A13: Sit down	A23: Swing objects	A33: Put on glasses	A43: Long jump
A4: Falling down	A14: Stand up	A24: Throw	A34: Take off glasses	A44: Push up
A5: Waving one hand	A15: Sit and stand	A25: Staggering	A35: Pick up	A45: Sit up
A6: Waving two hands	A16: Knead face	A26: Headache	A36: Put on bag	A46: Shake hands (two-players)
A7: Clap	A17: Nod head	A27: Stomachache	A37: Take off bag	A47: Fighting (two-players)
A8: Rub hands	A18: Shake head	A28: Back pain	A38: Put object into bag	A48: Handing objects (two-players)
A9: Punch	A19: Thumb up	A29: Vomit	A39: Take object out of bag	A49: Lifting chairs (two-players)

adjusted to a fixed orientation to ensure that the performer can be centred in each view of the camera. Compared to the previous THU$^{\text{E-ACT}}$-50 which only contains 2 full front viewpoints (e.g., Camera #2 and Camera #3), the THU$^{\text{MV-EACT}}$-50 includes additional 2 side-frontal viewpoints and 2 backward viewpoints, making this dataset suitable for multi-view action recognition. Due to data transmission bandwidth limitations, every two event cameras are connected to a laptop for acquisition. Synchronous triggers ensure simultaneous start and stop of recording across all 6 cameras. The THU$^{\text{MV-EACT}}$-50 dataset contains 105 socially recruited subjects, covering all age groups of males and females (15-72 years), which is consistent with the previous THU$^{\text{E-ACT}}$-50. Fig. 5 displays some sampled sequences from all viewpoints. The average duration of each recording across all action categories is 2.34 seconds. Compared with the existing dataset, the THU$^{\text{MV-EACT}}$-50 dataset has a total of 31,500 video recordings in 6 perspectives, aiming to provide data support for academic research and application of the multi-view event-based action recognition task.

V. EXPERIMENTS AND RESULTS

A. Experimental Setup

Since only the DHP19 [18] and the collected THU$^{\text{MV-EACT}}$-50 datasets contain multi-view data, experiments are conducted on these two datasets to verify the effectiveness of the proposed method. In both the DHP19 and THU$^{\text{MV-EACT}}$-50 datasets, Top-1, Top-3, and Top-5 accuracy are used as evaluation metrics. In this paper, we have performed experiments in two settings: 1) cross-subject setting, where all viewpoints are input simultaneously during training, allowing us to evaluate the performance of the proposed method using disjoint training and validation or test sets from different performers; and 2) cross-view setting, where the training and test sets are divided based on the viewpoint numbers to assess the generalization ability of the proposed method to unseen viewpoints.

For the cross-subject experiments, we divide the training set, validation set and test set in the ratio of 8:1:1. Specifically, the DHP19 dataset is divided into 12 training objects, 2 validation objects, and 3 test objects, while the THU$^{\text{MV-EACT}}$-50 dataset has 85 subjects for training, 10 subjects for validation, and 10 subjects for testing. Since both datasets contain multiple views, they serve as suitable benchmarks for evaluating multi-view action recognition using event cameras. Regarding the cross-view experiments, we adopt a distinct approach. For the THU$^{\text{MV-EACT}}$-50 dataset, 4 views are used for training, 1 view for validation and 1 view for testing. As for the DHP19 dataset, 3 views are employed for training, and 1 view is used for testing. Notably, both the training and test sets encompass all subjects, ensuring comprehensive coverage.

B. Implementation Details

The key aspect of the proposed method lies in utilizing hypergraph neural networks to capture high-order associations among features across different viewpoints and temporal segments. To demonstrate the effectiveness of each proposed module, we establish a multi-view baseline method that directly fuses these features after converting the raw event data into a frame-like representation and passing them through the view feature extraction module during training. Notably, the multi-view baseline approach does not apply the hypergraph neural network and the vertex attention mechanism. Moreover, we also investigate

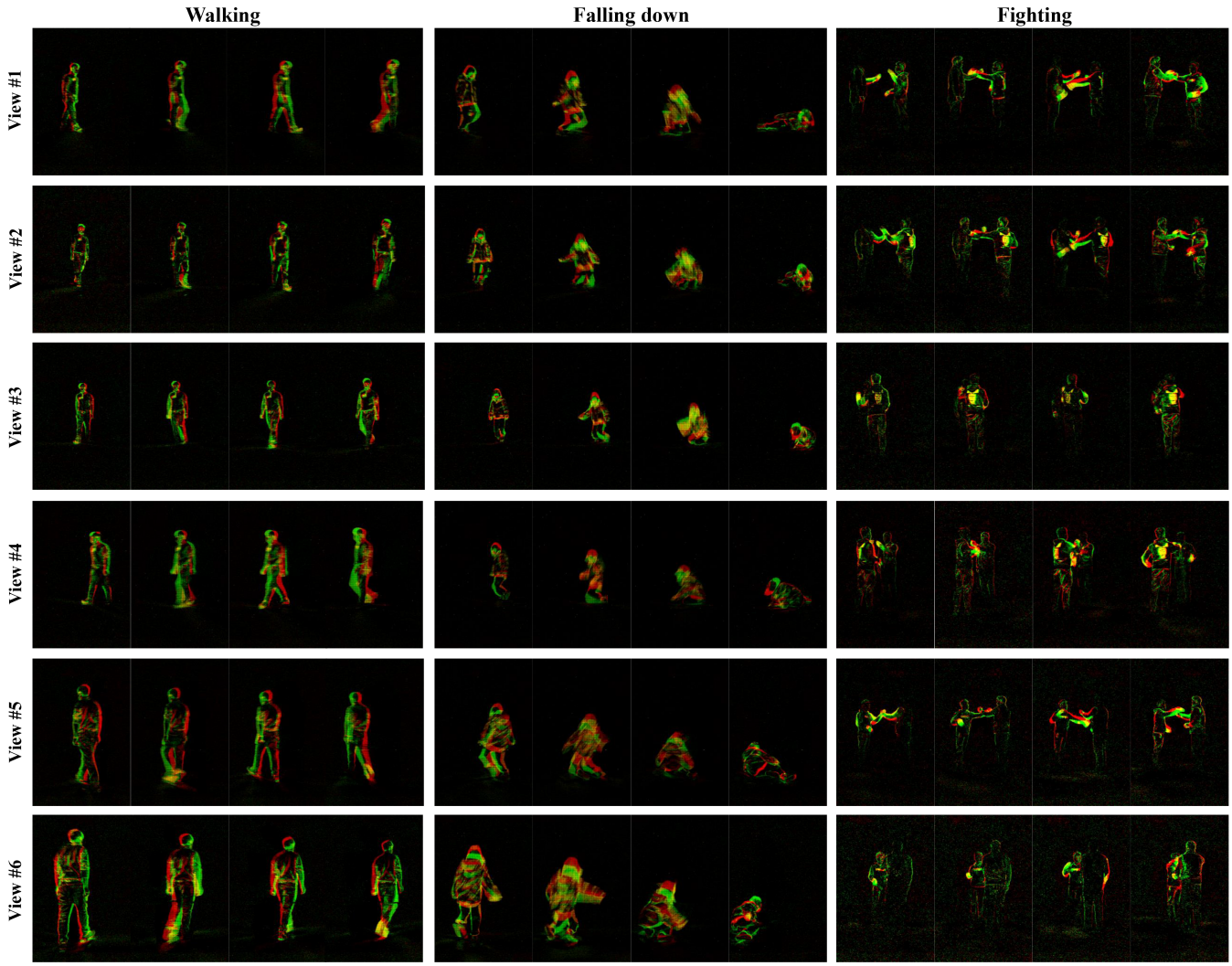

Fig. 5. Examples in the collected THU$^{\text{MV-EACT}}$-50 dataset.

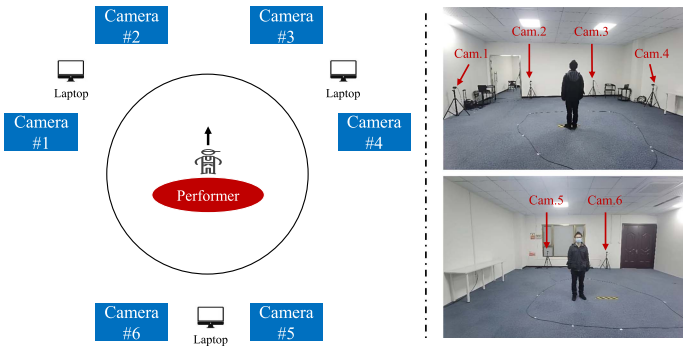

Fig. 6. Data acquisition environment of the collected THU$^{\text{MV-EACT}}$-50 dataset.

the application of graph neural network structure and the direct utilization of a single-view baseline network, both of which are elaborated upon in subsequent subsections.

For the main experiments, ResNet 18 [87] pre-trained on ImageNet [88] is used as the backbone network to meet the low-power requirements of event processing. The number of time windows T in event processing is set to 9, the number of k in the KNN-based hyperedge is set to 3, and the number of L in hypergraph propagation is set to 2. The network has been trained for 40 epochs on all benchmarks using the Adam [89] optimizer with an initial learning rate value of 1×10^{-4}, a weight decay of 1×10^{-4} and a batch size of 12. The exponential learning rate decay [90] strategy is applied, with a gamma of 0.5. All experiments are implemented based on PyTorch [91], and training with a Tesla V100 GPU.

C. Quantitative Results

1) Cross-Subject Evaluation: In the cross-subject experiments, we divide the training, validation and test set based on the number of subjects, and simultaneously input data from all views of a given sample for action classification in the training phase. We set up two baselines for comparison: a single-view and a multi-view, both of which transform the raw event data into frame-like representations. In the training phase, the single-view baseline only inputs one view at a time, while the multi-view baseline employs the view feature extraction module to input

TABLE III
COMPARATIVE ANALYSIS OF THE ACCURACY OF SINGLE-VIEW, MULTI-VIEW BASELINES, AND THE PROPOSED METHOD ACROSS CROSS-SUBJECT AND CROSS-VIEW SCENES

Type	Method	Cross-subject						Cross-view					
		DHP19 [18]			$THU^{MV-EACT}$-50			DHP19 [18]			$THU^{MV-EACT}$-50		
		Top-1	Top-3	Top-5	Top-1	Top-3	Top-5	Top-1	Top-3	Top-5	Top-1	Top-3	Top-5
Single-view	Baseline	78.94	94.26	99.38	89.98	96.85	98.34	30.63	56.62	70.08	47.98	66.95	77.82
	EV-ACT [16]	82.64	95.31	99.58	92.15	97.53	98.74	37.25	63.23	75.34	51.26	70.13	77.69
Multi-view	Baseline	81.43	95.20	99.10	92.53	97.59	98.84	39.28	65.47	76.63	54.38	73.46	80.39
	HyperMV-GNN	85.70	97.55	99.63	94.82	98.47	99.88	47.62	66.61	77.95	56.43	74.26	82.13
	HyperMV	**92.42**	**98.65**	**100**	**95.74**	**99.23**	**99.90**	**51.63**	**67.18**	**78.49**	**58.54**	**78.07**	**83.92**

multiple views and concatenates the features obtained from each view for action classification. Further, we also construct a GNN-based method named "HyperMV-GNN" on top of the multi-view baseline. For the rule-based strategy, it connects features from adjacent time sequences within the same view and connects features from different views within the same time. In the knn-based strategy, each vertex is connected to its k most similar vertices via k vertex-to-vertex edges. The graph convolution operation and the vertex attention mechanism are utilized to fuse features. To encapsulate our entire model, the proposed complete framework based on Hypergraph Neural Network (HGNN) is referred to as "HyperMV".

Table III presents the recognition accuracies of various methods on the DHP19 and $THU^{MV-EACT}$-50 datasets under the cross-subject setting. In addition to the single-view baseline, we also employ EV-ACT [16], the current SOTA method of single-view action recognition for comparison. When compared to the single-view baseline, the multi-view baseline improves Top-1 accuracy by 2.49% and 2.55% on the DHP19 and $THU^{MV-EACT}$-50 datasets, respectively. Even comparing to the single-view SOTA method, the multi-view baseline can approach or even exceed the accuracy of EV-ACT [16], demonstrating the advantages of using information from multiple viewpoints. Furthermore, the GNN and HGNN-based methods achieve higher accuracy compared to the multi-view baseline, thanks to their superior fusion of features from different viewpoints and temporal sequences. Specifically, HyperMV-GNN enhances Top-1 accuracy by 4.27% and 2.29% on the two datasets. Meanwhile, HyperMV can boost Top-1 accuracy by 10.99% and 3.21%, respectively. Due to the hypergraph's capability to model high-order correlations (e.g., feature fusion from the same viewpoint at any moment via 1-hop), HyperMV holds an advantage over HyperMV-GNN, particularly on small datasets like DHP19.

2) Cross-View Evaluation: Cross-view evaluation aims to test the model's generalization capacity for unseen views. Specifically, for the DHP19 dataset, we use 3 viewpoints for training and 1 for testing, while for the $THU^{MV-EACT}$-50 dataset, 4 viewpoints are used for training, 1 viewpoint for validation and 1 viewpoint for testing. As demonstrated in Table III, all baselines and methods encountered significant accuracy degradation compared to the cross-subject setting. The single-viewpoint baseline suffers a decrease in Top-1 accuracy by 48.3% and 42.0% on the DHP19 and $THU^{MV-EACT}$-50 datasets, respectively. Comparatively, the multi-view baseline enhances Top-1 accuracy by 8.65% and 6.40% on the two datasets due to its ability to explore feature associations between different views during training. The GNN and HGNN-based methods proposed in our paper demonstrate stronger cross-view generalization capacities. In particular, HyperMV-GNN further improves Top-1 accuracy by 8.34% and 2.05% over the multi-view baseline on both datasets. Meanwhile, HyperMV enhances Top-1 accuracy by 12.35% and 4.16% compared to the multi-view baseline. The more pronounced performance improvement on the DHP19 dataset compared to the $THU^{MV-EACT}$-50 dataset can primarily be attributed to the fact that DHP19 encompasses only four views. This condition makes the proposed GNN and HGNN-based method more effective in augmenting model generalization across views, particularly when handling complex association modeling.

TABLE IV
COMPARISON WITH SOTA OF FRAME-BASED MULTI-VIEW ACTION RECOGNITION IN TERMS OF TOP-1 ACCURACY

Method	Cross-subject		Cross-view	
	DHP19	$THU^{MV-EACT}$-50	DHP19	$THU^{MV-EACT}$-50
CNN-BiLSTM [83]	73.43	84.34	30.05	39.90
Att-LSTM [92]	76.03	86.32	34.10	41.29
DA-NET [9]	84.07	92.85	43.58	51.10
CVAction [10]	85.46	93.52	46.42	54.26
ViewCLR [93]	90.28	94.21	47.78	55.82
HyperMV	**92.42**	**95.74**	**51.63**	**58.54**

3) Comparisons With Frame-Based Methods: Since there exists no work on event-based multi-view action recognition, we compare the proposed HyperMV with several classical works in frame-based multi-view action recognition. Specifically, we view the event frames processed by Event Processing module as natural images, which are then fed into frame-based frameworks, including CNN-BiLSTM [83], Att-LSTM [92], DA-NET [9], CVAction [10], and ViewCLR [93]. As detailed in Table IV, the results show that HyperMV outperforms the state-of-the-art (SOTA) methods in both cross-subject and cross-view scenarios

TABLE V
COMPARATIVE ANALYSIS OF TOP-1 ACCURACY ACROSS DIFFERENT HYPERGRAPH CONSTRUCTION STRATEGIES

Strategy	Cross-subject		Cross-view	
	DHP19	THU$^{\text{MV-EACT}}$-50	DHP19	THU$^{\text{MV-EACT}}$-50
Rule-based	90.42	93.89	47.14	57.18
KNN-based	91.03	93.15	47.80	56.92
Rule-based + KNN-based	**92.42**	**95.74**	**51.63**	**58.54**

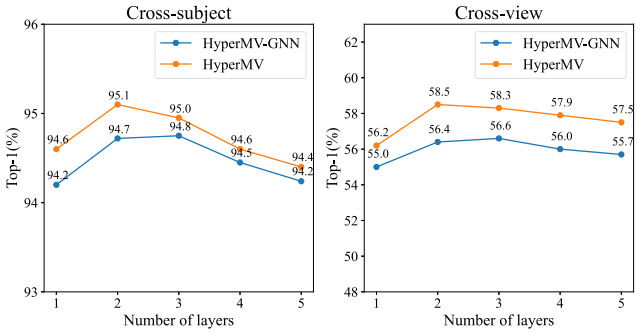

Fig. 7. Impact of varying hypergraph propagation layers on the Top-1 accuracy.

for both two datasets. Specifically, on the DHP19 dataset, HyperMV achieves a 2.14% and 5.21% increase in Top-1 accuracy over the SOTA's ViewCLR [93] in cross-subject and cross-view scenarios, respectively. Similarly, on the THU$^{\text{MV-EACT}}$-50 dataset, it shows improvements of 1.53% and 2.72% in two scenarios, respectively. These results indicate that HyperMV holds significant advantages in multi-view action recognition for event data. The reason mainly lies in the proposed multi-view hypergraph neural network, which effectively integrates features across various viewpoints and moments, alleviating the critical issues of information deficit and semantic misalignment often encountered in event-based multi-view action recognition.

D. Component Analysis

Hypergraph Construction Strategy: The proposed hypergraph construction strategy integrates both rule-based and KNN-based hyperedges. To evaluate the effectiveness of these different strategies, we perform experiments in three settings: purely rule-based, purely KNN-based, and a combination of both. All settings are based on the proposed multi-view hypergraph neural network. The results are outlined in Table V. The KNN-based strategy outperforms the rule-based strategy on the DHP19 dataset in both cross-subject and cross-view scenes. Conversely, the rule-based strategy yields better results on the THU$^{\text{MV-EACT}}$-50 dataset. However, the highest performance is achieved across all scenarios when both types are employed. Relative to the most effective individual strategy, the simultaneous use of both hyperedges improves the Top-1 accuracy by 1.39% and 1.85% on the DHP19 and THU$^{\text{MV-EACT}}$-50 datasets in the cross-subject setting, and by 3.83% and 1.36% in the cross-view setting. These findings suggest that explicit and implicit associations between perspectives and temporal sequences offer distinct benefits on different datasets. Simultaneously leveraging both strategies enables the system to draw from rule-based correlations as well as uncover implicit associations.

Number of Hypergraph Layers: To investigate the impact of the number of hypergraph convolutional layers on multi-view event-based action recognition performance, we conduct experiments using two network structures, i.e., HyperMV-GNN and HyperMV, with varying numbers of layers. For the THU$^{\text{MV-EACT}}$ dataset, we perform feature fusion across views and temporal segments under both cross-subject and cross-view scenes, with the number of layers L ranging from 1 to 5. The results are illustrated in Fig. 7. According to the experimental outcomes, HyperMV consistently outperforms the HyperMV-GNN approach, which suggests that HGNN possess superior capabilities in establishing feature correlation. As for the varying number of layers, alterations in the number of graph and hypergraph layers can influence recognition accuracy. In the HyperMV-GNN, the best Top-1 accuracy in both scenarios is achieved with $L = 3$, obtaining 94.8% under the cross-subject and 56.6% under the cross-view setting. Given that HyperMV considers more complex correlations, it achieves optimal performance with $L = 2$, yielding Top-1 accuracies of 95.1% and 58.5% respectively. However, both HyperMV-GNN and HyperMV experience a decrease when the number of layers continues to increase. This might be attributed to the "over-smoothing" phenomenon caused by excessive feature fusion, in which vertex features converge through overly extensive propagation, thereby diminishing the network's ability to capture locally differentiated features. Hence, it is advisable to set the number of graph and hypergraph convolution layers within the range of $L \in [2, 3]$.

TABLE VI
TOP-1 ACCURACY ON THE THU$^{\text{MV-EACT}}$-50 DATASET WITH OR WITHOUT THE VERTEX ATTENTION MECHANISM

Method	Attention	Cross-subject	Cross-view
Multi-view Baseline	×	91.35	53.19
	✓	**92.53**	**54.38**
HyperMV-GNN	×	93.02	55.12
	✓	**94.82**	**56.43**
HyperMV	×	93.42	57.21
	✓	**95.74**	**58.54**

Vertex Attention Mechanism: The vertex attention mechanism in this paper encompasses both the proposed vertex attention hypergraph propagation and the final vertex weighting operator. To validate the efficacy of the vertex attention mechanism, we perform ablation experiments on the THU$^{\text{MV-EACT}}$-50 dataset with and without the use of the vertex attention mechanism. In addition to the experiments on HyperMV-GNN and HyperMV, we also test the effects of using the traditional attention weight (i.e., weighting the features obtained from V views) on the multi-view baseline. As shown in Table VI, the utilization of the attention mechanism yields an enhancement in Top-1 recognition accuracy in both cross-subject and cross-view settings. The vertex attention mechanism improves by 1.80% and 1.31% under HyperMV-GNN for cross-subject and cross-view scenarios,

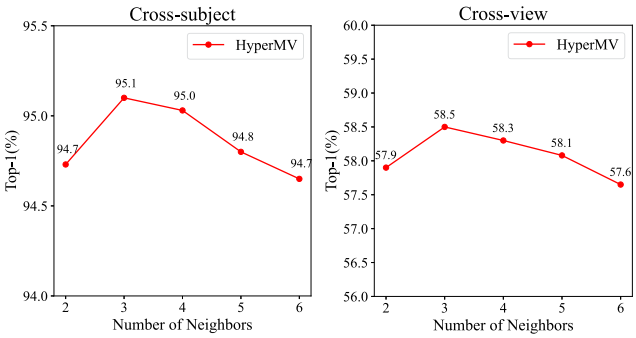

Fig. 8. Impact of varying number of neighbors in the KNN-based hyperedge on the Top-1 accuracy.

and by 2.32% and 1.33% under HyperMV, respectively. Additionally, the conventional attention mechanism also provides an increase in recognition accuracy under the multi-view baseline, improving by 1.18% and 1.19% in the two scenarios. These results indicate that there exist discrepancies in the significance of event features under different viewpoints and moments, and assigning attention weights to these features can result in better performance.

KNN-Based Hyperedge: The hypergraph construction approach proposed in this paper incorporates a KNN-based hyperedge strategy, which establishes hyperedge connections based on the K-nearest neighbor of vertex embedding. To assess the impact of the number of neighbors on the performance, we examine changes in Top-1 recognition accuracy by modifying the value of k on the THU$^{\text{MV-EACT}}$-50 dataset, as shown in Fig. 8. It can be observed that when k is small (e.g., $k = 2$), the limited quantity of neighbors hampers the hypergraph's capacity to model complex relationships among vertices, devolving into a standard GNN. In the majority of cases, a slight increase in k aids the model in capturing distant feature associations better, thereby improving recognition accuracy. For instance, when $k = 3$, the proposed method attains the highest Top-1 recognition accuracy in both cross-subject and cross-view settings (95.1% and 58.5%, respectively). However, a further increase of k results in a decline in accuracy. For example, when $k = 6$, the recognition accuracy drops to 94.7% and 57.6% respectively. This suggests that an excessively large number of K-nearest neighbors leads the model to over-fuse information with significant variations in viewpoint and temporal sequence. Consequently, the number of neighbors can have some slight effects on the performance of the multi-view event-based action recognition.

E. Model Complexity

To quantitatively assess the efficiency of the proposed HyperMV, we conduct experiments on the model complexity within the cross-subject test set of THU$^{\text{MV-EACT}}$-50. The model parameters (M) and the number of floating-point operations per second (GFlops) are employed as evaluation metrics. Three classical multi-view frame-based action recognition methods are selected for comparisons, including DA-NET [9], CVAction [10] and ViewCLR [93]. The results shown in Table VII indicate that the proposed framework requires only 22.8 M parameters and 31.2 GFlops per viewpoint, substantially undercuts the complexity of all frame-based counterparts. Among them, CVAction emerges as the most complex due to its reliance on a 3D CNN backbone. In contrast, HyperMV achieves better performance in multi-view event-based action recognition with a significantly lower complexity.

TABLE VII
COMPARISON OF EVENT-BASED AND FRAME-BASED MULTI-VIEW ACTION RECOGNITION IN TERMS OF MODEL COMPLEXITY

Method	Params (M)	Flops (G) × Views
DA-NET [9]	28.5	75.6 × 6
CVAction [10]	72.1	183.4 × 6
ViewCLR [93]	32.5	72.4 × 6
HyperMV	**22.8**	**31.2** × **6**

VI. CONCLUSION

In this paper, we introduce *HyperMV*, a pioneering framework for multi-view event-based action recognition. The proposed framework converts the inherently discrete event data into frame-like representations for each viewpoint. Through the implementation of the multi-view hypergraph neural network by employing rule-based and KNN-based strategies, the framework not only augments the capacity to capture explicit and implicit feature associations but also adds an extra dimension to the current landscape of multi-view action recognition research. Moreover, HyperMV incorporates a vertex attention mechanism to further enhance its action recognition efficacy. We also contribute to the broader research community by constructing the largest multi-view event action dataset, i.e., $\mathbf{THU^{MV-EACT}-50}$, which will serve as a significant resource for future academic evaluations and real-world applications.

REFERENCES

[1] J. Carreira and A. Zisserman, "Quo Vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6299–6308.
[2] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 510–526.
[3] D. P. Losey, K. Srinivasan, A. Mandlekar, A. Garg, and D. Sadigh, "Controlling assistive robots with learned latent actions," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 378–384.
[4] T. Kosch, J. Karolus, J. Zagermann, H. Reiterer, A. Schmidt, and P. W. Woźniak, "A survey on measuring cognitive workload in human-computer interaction," *ACM Comput. Surv.*, vol. 55, 2023, Art. no. 283.
[5] Y. Kong and Y. Fu, "Human action recognition and prediction: A survey," *Int. J. Comput. Vis.*, vol. 130, no. 5, pp. 1366–1401, 2022.
[6] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," *Int. J. Comput. Vis.*, vol. 119, no. 3, pp. 219–238, 2016.
[7] B. Jiang, M. Wang, W. Gan, W. Wu, and J. Yan, "STM: Spatiotemporal and motion encoding for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2000–2009.
[8] H. Duan, Y. Zhao, K. Chen, D. Lin, and B. Dai, "Revisiting skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2969–2978.
[9] D. Wang, W. Ouyang, W. Li, and D. Xu, "Dividing and aggregating network for multi-view action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 451–467.

[10] S. Vyas, Y. S. Rawat, and M. Shah, "Multi-view action recognition using cross-view video prediction," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 427–444.

[11] T. Delbruck, "Neuromorphic vision sensing and processing," in *Proc. Eur. Solid-state Device Res. Conf.*, 2016, pp. 7–14.

[12] R. Berner, C. Brandli, M. Yang, S.-C. Liu, and T. Delbruck, "A 240× 180 10mw 12us latency sparse-output vision sensor for mobile applications," in *Proc. Symp. VLSI Circuits*, 2013, pp. 186–187.

[13] R. Xiao, H. Tang, Y. Ma, R. Yan, and G. Orchard, "An event-driven categorization model for AER image sensors using multispike encoding and learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3649–3657, Sep. 2020.

[14] Q. Liu, D. Xing, H. Tang, D. Ma, and G. Pan, "Event-based action recognition using motion information and spiking neural networks," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 1743–1749.

[15] J. Chen, J. Meng, X. Wang, and J. Yuan, "Dynamic graph CNN for event-camera based gesture recognition," in *Proc. Int. Symp. Circuits Syst.*, 2020, pp. 1–5.

[16] Y. Gao et al., "Action recognition and benchmark using event cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 14081–14097, Dec. 2023.

[17] S. Miao et al., "Neuromorphic vision datasets for pedestrian detection, action recognition, and fall detection," *Front. Neurorobot.*, vol. 13, pp. 38–45, 2019.

[18] E. Calabrese et al., "DHP19: Dynamic vision sensor 3D human pose dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1695–1704.

[19] B. R. Pradhan, Y. Bethi, S. Narayanan, A. Chakraborty, and C. S. Thakur, "N-HAR: A neuromorphic event-based human activity recognition system using memory surfaces," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2019, pp. 1–5.

[20] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang, "Optical flow guided feature: A fast and robust motion representation for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1390–1399.

[21] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4305–4314.

[22] J. Lin, C. Gan, and S. Han, "TSM: Temporal shift module for efficient video understanding," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 7083–7093.

[23] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.

[24] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4597–4605.

[25] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[26] L. Wang et al., "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 20–36.

[27] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1933–1941.

[28] Y. Wang, J. Song, L. Wang, L. Van Gool, and O. Hilliges, "Two-stream SR-CNNs for action recognition in videos," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 108.1–108.12.

[29] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6202–6211.

[30] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa, "Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2273–2286, Nov. 2011.

[31] J. Zheng and Z. Jiang, "Learning view-invariant sparse representations for cross-view action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3176–3183.

[32] L. Liu and L. Shao, "Learning discriminative representations from RGB-D video data," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 1493–1500.

[33] Y. Kong and Y. Fu, "Bilinear heterogeneous information machine for RGB-D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1054–1062.

[34] F. Nie et al., "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 1881–1887.

[35] A. Ullah, K. Muhammad, T. Hussain, and S. W. Baik, "Conflux LSTMs network: A novel approach for multi-view action recognition," *Neurocomputing*, vol. 435, pp. 321–329, 2021.

[36] Y. Bai, Z. Tao, L. Wang, S. Li, Y. Yin, and Y. Fu, "Collaborative attention mechanism for multi-view action recognition," 2020, arXiv: *2009.06599*.

[37] K. Shah, A. Shah, C. P. Lau, C. M. de Melo, and R. Chellappa, "Multi-view action recognition using contrastive learning," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2023, pp. 3381–3391.

[38] L. Wang, Z. Ding, Z. Tao, Y. Liu, and Y. Fu, "Generative multi-view human action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6211–6220.

[39] Z. Liang, M. Yin, J. Gao, Y. He, and W. Huang, "View knowledge transfer network for multi-view action recognition," *Image Vis. Comput.*, vol. 118, 2022, Art. no. 104357.

[40] R. J. Walker, J. A. Richardson, and R. K. Henderson, "A 128 × 96 pixel event-driven phase-domain -based fully digital 3D camera in 0.13 m CMOS imaging technology," in *Proc. IEEE Int. Solid-State Circuits Conf.*, 2011, pp. 410–412.

[41] R. Ghosh, A. Gupta, A. Nakagawa, A. Soares, and N. Thakor, "Spatiotemporal filtering for event-based action recognition," 2019, arXiv: *1903.07067*.

[42] S. U. Innocenti, F. Becattini, F. Pernici, and A. Del Bimbo, "Temporal binary representation for event-based action recognition," in *Proc. Int. Conf. Pattern Recognit.*, 2021, pp. 10426–10432.

[43] X. Wu and J. Yuan, "Multipath event-based network for low-power human action recognition," in *Proc. World Forum Internet Things*, 2020, pp. 1–5.

[44] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2005, pp. 729–734.

[45] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[46] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[47] D. K. Duvenaud et al., "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2224–2232.

[48] S. I. Ktena et al., "Distance metric learning using graph convolutional networks: Application to functional brain networks," in *Proc. Med. Image Comput. Comput.- Assist. Intervention*, 2017, pp. 469–477.

[49] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du, and C. Zou, "Hypergraph learning: Methods and practices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2548–2566, May 2022.

[50] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, Art. no. 544.

[51] J. B. Lee, R. Rossi, and X. Kong, "Graph classification using structural attention," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1666–1674.

[52] X. Han, Z. Jiang, N. Liu, and X. Hu, "G-mixup: Graph data augmentation for graph classification," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 8230–8248.

[53] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu, "Learning deep representations for graph clustering," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1293–1299.

[54] Y. Hu et al., "Adaptive hypergraph auto-encoder for relational data clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2231–2242, Mar. 2023.

[55] E. Müller, "Graph clustering with graph neural networks," *J. Mach. Learn. Res.*, vol. 24, pp. 1–21, 2023.

[56] L. Cai and S. Ji, "A multi-scale approach for graph link prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 3308–3315.

[57] Z. Zhu, Z. Zhang, L.-P. Xhonneux, and J. Tang, "Neural Bellman-Ford networks: A general graph neural network framework for link prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 29476–29490.

[58] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[59] A. Quek, Z. Wang, J. Zhang, and D. Feng, "Structural image classification with graph neural networks," in *Proc. Int. Conf. Digit. Image Comput. Techn. Appl.*, 2011, pp. 416–421.

[60] X. Zhou, Y. Zhang, and Q. Wei, "Few-shot fine-grained image classification via GNN," *Sensors*, vol. 22, no. 19, pp. 7640–7652, 2022.

[61] G. Liu and J. Wu, "Video-based person re-identification by intra-frame and inter-frame graph neural network," *Image Vis. Comput.*, vol. 106, 2021, Art. no. 104068.
[62] J. Lu, H. Wan, P. Li, X. Zhao, N. Ma, and Y. Gao, "Exploring high-order spatio-temporal correlations from skeleton for person re-identification," *IEEE Trans. Image Process.*, vol. 32, pp. 949–963, 2023.
[63] P. Wang, C. Yuan, W. Hu, B. Li, and Y. Zhang, "Graph based skeleton motion representation and similarity measurement for action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 370–385.
[64] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, Art. no. 912.
[65] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1601–1608.
[66] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 3558–3565.
[67] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. Int. Conf. Pattern Recognit.*, 2004, pp. 32–36.
[68] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas, "The i3DPost multi-view and 3D human action/interaction database," in *Proc. Conf. Vis. Media Prod.*, 2009, pp. 159–168.
[69] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, arXiv: *1212.0402*.
[70] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, "Cross-view action modeling, learning and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2649–2656.
[71] J. Liu et al., "NTU: RGB D 120: A large-scale benchmark for 3D human activity understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2684–2701, Oct. 2020.
[72] C. Liu, Y. Hu, Y. Li, S. Song, and J. Liu, "PKU-MMD: A large scale benchmark for skeleton-based human action understanding," in *Proc. Workshop Vis. Anal. Smart Connected Communities*, 2017, pp. 1–8.
[73] Y. Ji, F. Xu, Y. Yang, F. Shen, H. T. Shen, and W.-S. Zheng, "A large-scale RGB-D database for arbitrary-view human action recognition," in *Proc. ACM Int. Conf. Multimedia*, 2018, pp. 1510–1518.
[74] J. Jang, D. Kim, C. Park, M. Jang, J. Lee, and J. Kim, "ETRI-activity3D: A large-scale RGB-D dataset for robots to recognize daily activities of the elderly," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 10990–10997.
[75] C. Plizzari et al., "E2 (go) motion: Motion augmented event stream for egocentric action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 19935–19947.
[76] Y. Hu, H. Liu, M. Pfeiffer, and T. Delbruck, "DVS benchmark datasets for object tracking, action recognition, and object recognition," *Front. Neurosci.*, vol. 10, pp. 210251–210262, 2016.
[77] T. Serrano-Gotarredona and B. Linares-Barranco, "A 128 128 1.5% contrast sensitivity 0.9% FPN 3 s latency 4 mW asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers," *IEEE J. Solid-State Circuits*, vol. 48, no. 3, pp. 827–838, Mar. 2013.
[78] S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *Int. J. Neural Syst.*, vol. 19, no. 04, pp. 295–308, 2009.
[79] S. Li, Y. Feng, Y. Li, Y. Jiang, C. Zou, and Y. Gao, "Event stream super-resolution via spatiotemporal constraint learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 4480–4489.
[80] Y. Gao, S. Li, Y. Li, Y. Guo, and Q. Dai, "SuperFast: 200× video frame interpolation via event camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7764–7780, Jun. 2023.
[81] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 407–417, Feb. 2014.
[82] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5419–5427.
[83] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 989–997.
[84] M. Almatrafi, R. Baldwin, K. Aizawa, and K. Hirakawa, "Distance surface for event-based optical flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 7, pp. 1547–1556, Jul. 2020.
[85] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–12.
[86] S. Chen and M. Guo, "Live demonstration: CeleX-V: A 1 M pixel multi-mode event-based sensor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 1682–1683.
[87] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[88] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[89] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, arXiv: *1412.6980*.
[90] Z. Li and S. Arora, "An exponential learning rate schedule for deep learning," 2019, arXiv: *1910.07454*.
[91] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 8026–8037, 2019.
[92] P. Zhang, J. Xue, C. Lan, W. Zeng, Z. Gao, and N. Zheng, "Adding attentiveness to the neurons in recurrent neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 135–151.
[93] S. Das and M. S. Ryoo, "ViewCLR: Learning self-supervised video representation for unseen viewpoints," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2023, pp. 5573–5583.

Yue Gao (Senior Member, IEEE) received the BS degree from the Harbin Institute of Technology, Harbin, China, and the ME and PhD degrees from Tsinghua University, Beijing, China. He is currently an associate professor with the School of Software, Tsinghua University.

Jiaxuan Lu received the BE degree in computer science and technology from Shanghai University, Shanghai, China, and the ME degree from the School of Software, Tsinghua University, Beijing, China. He is currently working as a researcher with the Shanghai Artificial Intelligence Laboratory. His current research interests include visual understanding and action recognition.

Siqi Li received the BE degree from the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, in 2019. He is currently working toward the PhD degree with the School of Software, Tsinghua University, Beijing. His research interests include computer vision and machine learning.

Yipeng Li received the BS and MS degrees in electronic engineering from the Harbin Institute of Technology, and the PhD degree in electronic engineering from Tsinghua University, China, in 2003, 2005 and 2011, respectively. He is currently an associate researcher with the Department of Automation, Tsinghua University. His research interests include computer vision, vision-based navigation UAS and 3D reconstruction and perception of general scene.

Shaoyi Du (Member, IEEE) received the double bachelor's degrees in computational mathematics and computer science, in 2002, the MS degree in applied mathematics, in 2005, and the PhD degree in pattern recognition and intelligence system from Xi'an Jiaotong University, Xi'an, China, in 2009. He is also a professor with Xi'an Jiaotong University. His research interests include computer vision, machine learning, and pattern recognition.