A Dynamic GCN with Cross-Representation Distillation for Event-Based Learning

Yongjian Deng[1 4], **Hao Chen**[2*], **Youfu Li**[3]

[1]College of Computer Science, Beijing University of Technology
[2]School of Computer Science and Engineering, Southeast University
[3]Department of Mechanical Engineering, City University of Hong Kong
[4]Engineering Research Center of Intelligence Perception and Autonomous Control, Ministry of Education, Beijing, China
yjdeng@bjut.edu.cn, haochen303@seu.edu.cn, meyfli@cityu.edu.hk

Abstract

Recent advances in event-based research prioritize sparsity and temporal precision. Approaches learning sparse point-based representations through graph CNNs (GCN) become more popular. Yet, these graph techniques hold lower performance than their frame-based counterpart due to two issues: (i) Biased graph structures that don't properly incorporate varied attributes (such as semantics, and spatial and temporal signals) for each vertex, resulting in inaccurate graph representations. (ii) A shortage of robust pretrained models. Here we solve the first problem by proposing a new event-based GCN (EDGCN), with a dynamic aggregation module to integrate all attributes of vertices adaptively. To address the second problem, we introduce a novel learning framework called cross-representation distillation (CRD), which leverages the dense representation of events as a cross-representation auxiliary to provide additional supervision and prior knowledge for the event graph. This frame-to-graph distillation allows us to benefit from the large-scale priors provided by CNNs while still retaining the advantages of graph-based models. Extensive experiments show our model and learning framework are effective and generalize well across multiple vision tasks.

Introduction

Event cameras hold high dynamic range, low power consumption, and high temporal resolution while maintaining data non-redundancy advantages. However, the effective representation and learning of event data (as shown in Fig. 1) is still a topic of ongoing research.

Event data representation methods can be divided into two categories: **frame-based methods** that sacrifice data sparsity and motion precision to make event data compatible with pre-trained CNNs, and **point-based methods** that protect the sparsity and temporal structure of event streams. However, due to discrepancies among different event attributes (*i.e.,* spatial, temporal coordinates, and semantics) and limited labeled data, existing point-based methods suffer from biased graph representation or inadequate training.

To fully leverage the inherent advantages of event cameras, we aim to address two key problems of current point-based solutions: (1) *how to deal with diverse attributes (semantic, space distance, and temporal cues) to determine vertices' neighbors*. (2) *How to facilitate event graph learning without relying on additional data.*

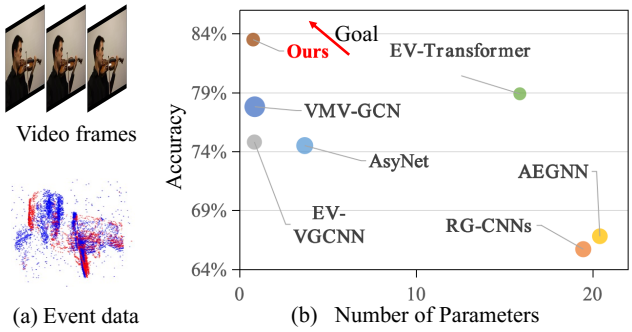
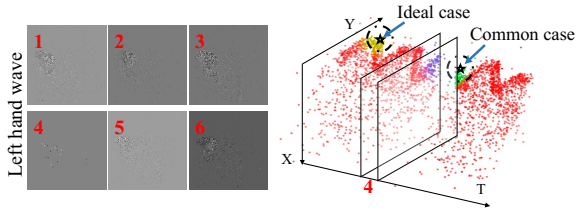


Figure 1: (a) Visual comparison between outputs from traditional cameras and event cameras. (b) Recognition accuracy vs model complexity (#Params) of our approach (EDGCN w/o and w/ CRD) on the N-Caltech dataset. The circle areas are proportional to the computational complexity (FLOPs).

Properly handling the neighborhood relations of vertices is key to achieving accurate event graph representations. Previous works in this direction usually follow traditional 3D vision methods and try to define neighborhoods by better handling spatio-temporal coordinates, such as unifying the value ranges of spatial-temporal coordinates (Bi et al. 2020; Deng et al. 2022; Li et al. 2021) or dynamically updating coordinates (Xie et al. 2022; Chen et al. 2020a). Fig. 2 shows that these methods have difficulty in effectively modeling discontinuous event streams caused by motion stagnation or occlusion. To this end, we introduce a simple yet effective graph construction strategy that define vertex neighborhoods considering all attributes in a learnable manner. We introduce an Event-based Dynamic Aggregation Layer (EDAL) that has multi-attribute joint learning branches for defining neighborhood integration strategies. We also include a rebalanced design that uses only coordinates for vertices' attentive aggregation, allowing the model to better judge the motion state of vertices.

To enhance the learning of event-based graph models, we borrow the valuable prior knowledge in well-pretrained CNNs to facilitate the event graph model. We propose

*Corresponding Author



(a) Frame-based representation (b) Point-based representation

Figure 2: Event-based representations for left-hand waving action. (a) Images integrated by events with different time intervals. Event stream (b) breaks in the fourth interval due to motion stagnation. Coordinate-defined neighborhoods can find temporally and semantically related neighbors for vertices ($\star$) in ideal cases. However, motion stagnation and occlusion problems make it hard to find highly related neighbors based on coordinates only (*e.g.,* purple vertex before motion stagnation). Thus, it is necessary to incorporate semantic information into neighborhood definition.

a cross-representation (frame-to-graph) distillation framework with a hybrid distillation structure, combining an intermediate feature-level contrastive loss (Chen et al. 2020b) and an inference-level distillation loss (Hinton, Vinyals, and Dean 2015; Romero et al. 2015). This framework transfers multi-level semantics and handles frame-graph discrepancies across layers. Compared to previous frame-to-frame transfer learning methods, *our method uses only event signals without extra data, while holding better learning and graph model advantages.* Our framework also shows better generalization despite the large representation discrepancy.

The main contributions are summarized as follows: (1) It is the first cross-representation distillation (frame-to-graph distillation) work for event data with no extra data required. We carefully analyze the variant discrepancy between frame and graph representation in different layers and use corresponding constraints to distill different layers. (2) We introduce a graph construction strategy with customized learning model for events, where the cross-vertex dependency is determined by the joint representation from all attributes of vertices in an unbiased and dynamic way. (3) Extensive experiments validate the efficacy of our proposed event graph and learning strategy on various downstream tasks, verifying the high generalization ability of our model.

Related Work

Event-Based Learning. Frame-based methods in event-based processing integrate events into dense representations, adapting them to CNNs for tasks like event-based recognition (Deng, Li, and Chen 2020; Deng, Chen, and Li 2021; Baldwin et al. 2022), video reconstruction (Rebecq et al. 2019; Zhu et al. 2022) and optical flow estimation (Hu et al. 2022). However, these methods sacrifice the sparsity and temporal precision of event data (Mitrokhin et al. 2020; Schaefer, Gehrig, and Scaramuzza 2022), resulting in redundant computation and high model complexity and quick inference, exploit event sparsity. Initially, Spiking Neural Networks (SNNs) (Orchard et al. 2015; Sironi et al. 2018) were used for event data due to their sparse and asynchronous nature. However, their training proved challenging. Combining SNNs with CNNs has been an approach (Wu et al. 2022; Yao et al. 2021) to solve this, but it introduces more computations. Then, PointNet-like models (Wang et al. 2019b; Sekikawa, Hara, and Saito 2019) are developed but face challenges in adaptively aggregating local features. Graph-based approaches (Mitrokhin et al. 2020; Bi et al. 2020; Li et al. 2021; Schaefer, Gehrig, and Scaramuzza 2022; Chen et al. 2020a) emerge to tackle the issue of defining neighborhoods based on event properties, proving to be more lightweight than frame-based methods with promising results in various applications. Current research has introduced new representations (Deng et al. 2022; Xie et al. 2022) or investigated spatial proximity of events (Li, Asif, and Ma 2022) to close the performance gap to frame-based methods. However, unbiased neighborhood definitions and graph model learning with limited labels remain unexplored. In this work, we propose a dynamic joint representation learning approach (EDGCN) with a cross-representation distillation training framework (CRD) to address both problems.

Transfer Learning on Event Data. Transfer learning has been extensively explored to ease the training of event-based models. Several studies (Rebecq et al. 2019; Tulyakov et al. 2022) transform events into RGB images to align with pretrained CNNs. Yet, these conversions increase computational costs. Cross-modality transfer approaches, as presented in (Hu, Delbruck, and Liu 2020; Deng et al. 2021; Sun et al. 2022; Messikommer et al. 2022), introduce extra supervision to aid event-based learning, but these additional visual modalities aren't consistently available. Some research (Gehrig et al. 2020; Rebecq, Gehrig, and Scaramuzza 2018) supplements event datasets by simulating data from traditional videos, but this often results in artifacts. Contrarily, our cross-representation distillation method taps into the shared knowledge from various deep models trained on diverse event data representations, eliminating the need for extra data. This method broadens applicability and enhances the learning of the event graph branch via a frame-to-graph distillation loss.

Approach

In this work, we devise a novel event-based dynamic graph CNN (EDGCN) and a cross-representation distillation strategy (CRD) to boost its learning sufficiency further. The pipeline of our proposed method is illustrated in Fig. 3, which contains the following key steps: (1) Representation construction of events for both the graph model and traditional CNNs. (2) Gradual aggregation of contexts using successive event-based dynamic aggregation layers (EDALs). (3) Parallel to the event graph branch, we map the original events to a frame-based branch as the teacher to promote EDGCN branch learning via the proposed cross-representation distillation framework. (4) The EDGCN is appended with different inference heads for various tasks. The following sections will clarify detailed designs and im-

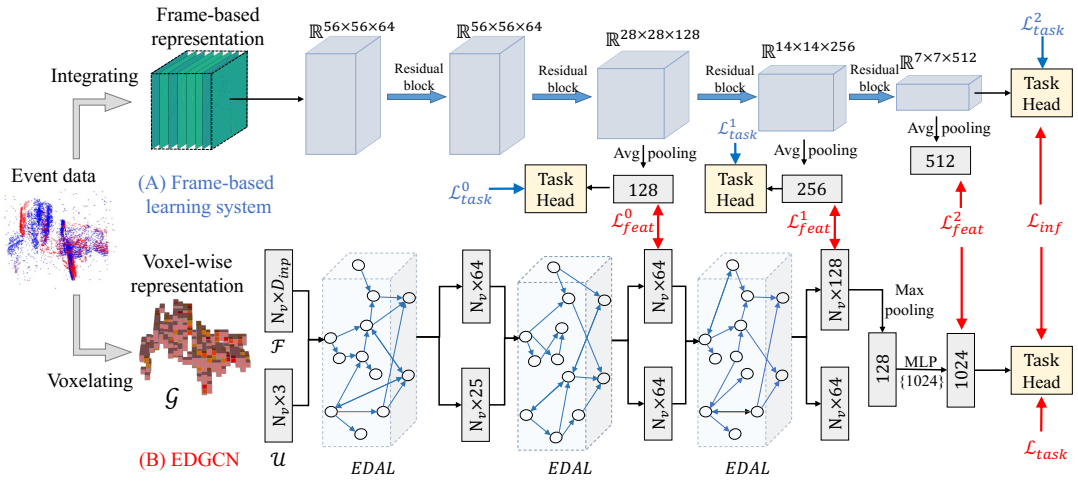


Figure 3: The pipeline of our proposed method takes two different representations of events as inputs for the frame-based model (A) and the point-based model (B), *i.e.,* EDGCN. A ResNet-like structure is chosen for the frame-based model. "MLP" stands for multi-layer perceptron, with numbers in brackets indicating layer sizes. Losses in blue are used for frame-based model training, while those in red are imposed when optimizing the EDGCN with CRD. *Only EDGCN is used at the inference stage.*

plementations in these steps.

Event-Based Representation

Each event e_i holds three properties: the occurred location ((x_i, y_i)), the triggered timestamp (t_i), and the polarity ($p_i \in \{-1, 1\}$). Particularly, the positive p denotes the brightness increase and vice versa. In this work, we directly adopt the voxel-wise representation (Deng et al. 2022) of events as our input. In specific, event streams ($\{e_i\}_N = \{x_i, y_i, t_i, p_i\}_N$) is firstly partitioned into voxels with the voxel size (v_x, v_y, v_t). Then, N_v voxels containing the largest number of events are reserved as vertices ($\mathcal{V}$) of the event-based graph ($\mathcal{G}$). Here, we denote the left-upper location of a vertex ($\mathcal{V}_i$) as its coordinate attribute ($\mathcal{U}_i = (x_i^v, y_i^v, t_i^v)$). Finally, the semantics ($\mathcal{F}_i \in \mathbb{R}^{D_{inp}}$) of the i-th vertex ($\mathcal{V}_i$) is obtained through event-wise integration formulated by the function Ω, where $D_{inp} = v_x v_y$.

EDGCN

We address the challenge of modeling dependencies of three attributes (spatial position, triggered time, and local semantics) in measuring cross-vertex edges with proposed graphs and dynamic aggregation layers. Our approach includes neighborhood definition, attentive aggregation, and coordinate attribute update. The dynamically updated graph improves neighborhood definition accuracy and feature aggregation efficacy by continuously refining vertex attributes.

Dynamic Aggregation Layer (EDAL). The main component of our EDGCN, namely EDAL, is schematized in Fig. 4. Its workflow is detailed successively as follows.

Neighborhood Definition. We suppose that the i-th vertex $\mathcal{V}_i$ is an input vertex to an EDAL with coordinate and semantic attributes $Attr(\mathcal{V}_i) : (\mathcal{U}_i \in \mathbb{R}^{D_u^{in}}, \mathcal{F}_i \in \mathbb{R}^{D_f^{in}})$. The goal of EDAL is to define neighborhood space for each vertex by considering all its attributes and aggregating attributes from its neighbors attentively. Considering the large discrepancy in spatial position, triggered time, and features of local semantics, we argue that projecting the $Attr(\mathcal{V}_i)$ to a unified feature space with Eq. 1 is required.

$$\mathcal{P}_i^F = \mathbb{M}^F(\mathcal{F}_i), \quad \mathcal{P}_i^U = \mathbb{M}^U(\mathcal{U}_i), \tag{1}$$

where $\mathbb{M}^F$ and $\mathbb{M}^U$ are *MLP*s for feature projection. The obtained representations $\mathcal{P}_i^F$ and $\mathcal{P}_i^U \in \mathbb{R}^{D_f^{in}}$ in the same feature space can then be fused as a joint representation $\mathcal{P}_i^{fuse}$ for vertex $\mathcal{V}_i$. Notably, we achieve the $\mathcal{P}_i^{fuse} \in \mathbb{R}^{D_f^{out}}$ through a fusion module ($\mathbb{F}$) consisting of an addition operation followed by a *MLP*. Next, we adopt the K-Nearest neighbor algorithm (*KNN*) on this joint representation to find the most relevant N_n neighbors of $\mathcal{V}_i$ in the $\mathcal{G}$. Here, we denote the set of edges between vertex $\mathcal{V}_i$ and its neighbors as $\mathcal{E}_i^{nei} \in \mathbb{R}^{N_n}$.

A direct approach to aggregating vertices' features is to follow the methods in (Veličković et al. 2018; Wang et al. 2019a), which achieve attentive aggregation by obtaining similarity or relative matrices of features (*i.e.,* $\mathcal{P}^{fuse}$) that define neighborhoods. However, there is one point that must be considered in our work. As shown in Fig. 2, in scenarios with complex motion states, vertices and their neighbors may be distant but semantically strongly related. We aim for the model to fully consider the motion correlation between neighbors and the central vertex when aggregating for such vertices rather than just global semantics. To this end, we enhance the contribution of coordinate attributes in graph representation for more accurate motion description by calculating aggregation weights using only coordinate clues. Further, we update the coordinate attributes with spatio-temporal relations in the neighborhood to enlarge their capability in motion description layer-by-layer. The ablations in Tab. 6 verify the efficacy of our intuitive design.

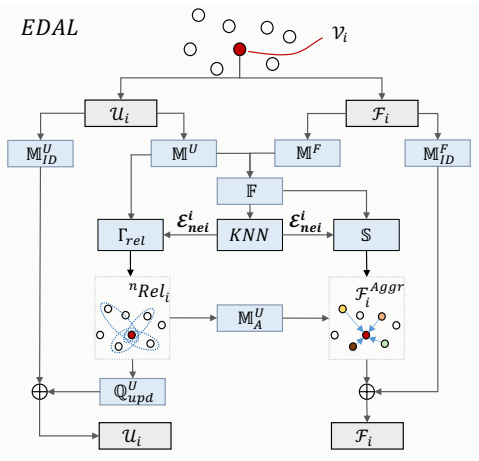


Figure 4: The detail structure of the EDAL. $\bigoplus$: element-wise addition.

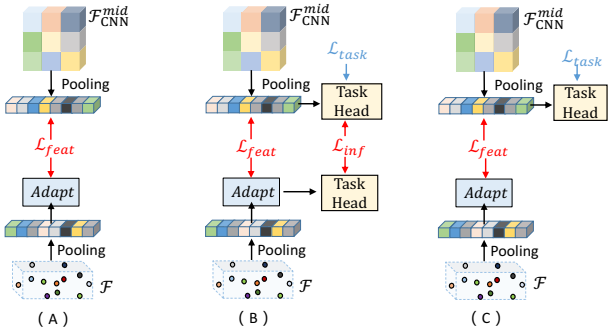


Figure 5: Different choices of distillation between intermediate features from frame-based models ($\mathcal{F}^{mid}_{CNN}$) and the EDGCN ($\mathcal{F}$). The adapt module is a *MLP* for dimension alignment between two feature vectors. We adopt (C) as our choice and detail the reason in the experiment section.

Attentive Aggregation. We calculate attention scores $w.r.t$ the coordinate attribute as formulated in Eq. 2.

$$Score_i = \mathbb{M}^U_A(\underset{j:(i,j)\in\mathcal{E}^{nei}_i}{\Gamma_{rel}}(\mathcal{P}^U_i, \mathcal{P}^U_j)), \tag{2}$$

where the function Γ_{rel} is used for concatenating its two inputs and stacking them under the constraint ($j : (i,j) \in \mathcal{E}^{nei}_i$). The function $\mathbb{M}^U_A$, a *MLP* with the Softmax activation, is imposed for mapping the input in $\mathbb{R}^{N_n \times 2D^{in}_f}$ to a vector of attentive scores $Score_i \in \mathbb{R}^{N_n}$. Next, we can aggregate features for vertex attentively as described in Eq. 3.

$$\mathcal{F}^{Aggr}_i = \sum(Score_i * (\underset{j:(i,j)\in\mathcal{E}^{nei}_i}{\mathbb{S}}(\mathcal{P}^{fuse}_j))), \tag{3}$$

where $\mathbb{S}$ works for stacking joint representations ($\mathcal{P}^{fuse}_j \in \mathbb{R}^{D^{out}_f}$) of all vertex's neighbors and its output is in $\mathbb{R}^{N_n \times D^{out}_f}$. The attentive scores $Score_i$ can then be applied to re-weight and obtain the aggregated features $\mathcal{F}^{Aggr}_i \in \mathcal{R}^{D^{out}_f}$ for vertex $\mathcal{V}_i$ through a summation operation.

Coordinate Attribute Update. We depict the derivation of updating coordinate attributes by $\mathcal{U}^{upd}_i = \mathbb{Q}^U_{upd}(Rel_i)$, where Rel_i is obtained in Eq. 2 via function Γ_{rel}, $\mathbb{Q}^U_{upd}$ consists of an average pooling followed by a *MLP* for feature aggregation and feature projection. After these two processes, the updated coordinate attribute $\mathcal{U}^{upd}_i \in \mathbb{R}^{D^{in}_f}$ of $\mathcal{V}_i$ can be achieved. By updating the coordinate attribute of each vertex with local spatio-temporal relations with its neighbors, the vertex is equipped with spatio-temporal positions and local motion associations simultaneously, which will be transmitted to the following layer.

Shortcut Connection. In the EDAL, two shortcut connections are included for both coordinate and feature attributes of vertices in the graph. In specific, two *MLP*s (M^U_{ID} and M^F_{ID}) are applied to input attributes $\mathcal{U}_i$ and $\mathcal{F}_i$ respectively. Finally, we add the attained features from M^U_{ID} and M^F_{ID} to our achieved updated coordinate attribute ($\mathcal{U}^{upd}_i$) and aggregated features ($\mathcal{F}^{Aggr}_i$) to obtain the final output of an EDAL, *i.e.*, $\mathcal{U}^{out}_i \in \mathbb{R}^{D^{in}_f}$ and $\mathcal{F}^{out}_i \in \mathbb{R}^{D^{out}_f}$.

Network Structure. The structure of the EDGCN is the same for all datasets except for the sub-stream task head. Three EDALs are cascaded to extract discriminative features from event data sequentially. For object recognition and action recognition tasks, we apply a Max pooling operation followed by a *MLP* after the third EDAL and then feed the output of this *MLP* to a fully connected layer for categorical prediction. As for the detection task, we follow the setting provided by (Schaefer, Gehrig, and Scaramuzza 2022) to apply a YOLO-based detection head (Redmon et al. 2016) to our extracted event-based contexts for object detection.

Cross-Representation Distillation (CRD)

As shown in Fig. 3, in our CRD framework, the teacher network is a frame-based learning branch with dense event-based representations as input and learning them starting from well-trained CNNs, while the student network is our *EDGCN* model. Then, a combined distillation constraint working on different layers is tailored to fulfill this transfer.

Distillation Framework. The key issue in achieving this challenging frame-to-graph transfer is the design of a distillation structure that carefully considers the varying cross-representation discrepancy across different layers. To this end, we propose a hybrid distillation approach that includes two views. (1) Inference-level distillation of contexts from final prediction outputs. For instance, we apply the distillation loss proposed in (Hinton, Vinyals, and Dean 2015) for classification tasks and the L1 loss (Romero et al. 2015) for regression-based tasks like position estimating in object detection. (2) Feature-level constraints that transfer hints from intermediate features of frame-based models. The knowledge transfer of intermediate features has been verified to effectively improve the training effect of the model to be transferred. However, for traditional CNNs and our EDGCN, their learning logic for event data from shallow to deep

Datasets	N-Cal	CIF10	N-C	DVS128
(v_x, v_y)	(10, 10)	(10, 10)	(5, 5)	(5, 5)
v_t	25 ms	60 ms	25 ms	40 ms
N_v	2048	2048	512	512
N_n	20	20	20	20

Table 1: Parameter settings of different datasets.

might be totally different. Thus, hard constraints (*e.g.,* L1/L2 distance) (Isola et al. 2017) would be too strict for our transfer task. For this reason, we exploit the contrastive loss, NT-Xent (Chen et al. 2020b; Huang et al. 2021), to realize the transfer by increasing the correlation between intermediate features from both networks. In specific, three variants to equip NT-Xent loss in the CRD are proposed in Fig. 5, with method Fig. 5.(C) being adopted as the final choice.

Optimization. The whole training process of our study can be divided into two parts. First, the training process of frame-based models with task-specific loss solely is described by $\mathcal{L}_{total}^{frame} = \sum_i^{N_t} \mathcal{L}_{task}^i$, where $\mathcal{L}_{task}^i$ are task-specific losses. They are applied to multiple prediction layers individually (Fig. 3). N_t represents the number of intermediate features used for cross-representation learning. Next, we optimize our proposed EDGCN with the loss for task-specific supervision ($\mathcal{L}_{task}$), the loss for inference-level knowledge transferring ($\mathcal{L}_{inf}$) and a series of contrastive losses ($\mathcal{L}_{feat}^i$) for feature-level distillation as described in Eq. 4.

$$\mathcal{L}_{total}^{Edgcn} = \lambda \mathcal{L}_{task} + (1-\lambda)\mathcal{L}_{inf} + \sum_i^{N_t} \mathcal{L}_{feat}^i, \quad (4)$$

where λ control the contribution of the first two components in the training process and set it as 0.5 for all experiments.

Experimental Results

We evaluate our proposed method on multiple tasks. Besides, we validate the superiority of the EDGCN on the model complexity (trainable parameters) and the number of floating-point operations (FLOPs). We compare our approach to representative methods from both frame-based (EST (Gehrig et al. 2019), YOLE (Cannici et al. 2019), M-LSTM, MVF-Net (Deng, Chen, and Li 2021), Asynet (Messikommer et al. 2020), LIAF-Net (Wu et al. 2022), TA-SNN (Yao et al. 2021)) and point-based (EventNet (Sekikawa, Hara, and Saito 2019), RG-CNNs (Bi et al. 2020), Evs-S (Li et al. 2021), EV-VGCNN (Deng et al. 2022), AEGNN (Schaefer, Gehrig, and Scaramuzza 2022), VMV-GCN (Xie et al. 2022)) branches. Finally, the efficacy of the proposed CRD and its generalizability are validated.

Implementation Details. We choose ResNet as the backbone of frame-based models used for applying CRD to object classification and object detection tasks and adopt I3D-R (w/ ResNet50) (Chen et al. 2021) for the action recognition task. We train them using the Adam optimizer with batch size 32 and an initial learning rate (*lr*) of 1e-4, which is reduced by a factor of 2 after 20 epochs. The dense input of these frame-based models is VoxelGrid. For the EDGCN, we keep its network structure (Fig. 3) unchanged for all datasets except for its task head. We use SGD optimizer with an initial *lr* of 1e-1 for object classification and action recognition, and reduce the *lr* until 1e-4 using cosine annealing. We choose Adam optimizer with batch size 32 for detection, and reduce *lr* starting from 1e-2 by a factor of 2 after 20 epochs. The settings are consistent for training the EDGCN solely and with CRD. We list the statistics of adopted datasets and their settings in Tab. 1. We average over five runs as our final results for all experiments.

Method	Type[§]	N-Cal	N-C	CIF10
Pretrained on ImageNet				
EST	F	0.837	0.925	0.749
M-LSTM	F	0.857	0.957	0.730
MVF-Net	F	**0.871**	**0.968**	**0.762**
Without pretraining				
EST	F	0.753	0.919	0.634
M-LSTM	F	0.738	0.927	0.631
MVF-Net	F	0.687	0.927	0.599
EventNet	P	0.425	0.750	0.171
RG-CNNs	P	0.657	0.914	0.540
EvS-S	P	0.761	0.931	0.680
EV-VGCNN	P	0.748	0.953	0.670
AEGNN	P	0.668	0.945	-
VMV-GCN	P	0.778	0.932	0.690
EV-Transformer	P	0.789	0.954	0.709
Ours[†]	P	**0.801**	**0.958**	**0.716**
Ours w/ CRD[‡]	P	**0.835**	**0.963**	**0.752**

Table 2: Comparison of models $w.r.t$ classification accuracy. †: *Performance of the EDGCN trained solely.* ‡: *Performance of the model trained with CRD.* § : *F:frame-based method; P:point-based method.*

Object Classification

Event-based object classification is an essential application since event cameras can recognize objects more accurately than traditional cameras in scenarios with severe motion blur and extreme lighting conditions. In this work, we select three challenging datasets commonly used for evaluating event-based object classification, *i.e.,* N-Cal (Orchard et al. 2015), N-C (Sironi et al. 2018), and CIF10 (Li et al. 2017) (Tab. 1). The ResNet-18 is the backbone of the frame-based model that we employed for optimizing EDGCN with CRD, and its performance on N-Cal, N-C, and CIF10 is 0.868, 0.964, and 0.757. The cross-entropy loss is utilized as $\mathcal{L}_{task}$ for the model's training.

Classification Accuracy. We compare the proposed method with SOTA methods falling in both point-based and frame-based categories. The Tab. 2 presents that methods with graph-based learning (RG-CNNs, Evs-S, EV-VGCNN, AEGNN, VMV-GCN, EV-Transformer and ours) are prevalent $w.r.t$ classification accuracy over other point-based methods. Notably, the proposed EDGCN achieves top performance among these graph-based approaches, revealing

Method	Type	Accuracy	GFLOPs	#Params
LIAF-Net	F	0.976	13.6	-
TA-SNN	F	**0.986**	-	-
RG-CNN (Res.3D)	P	0.972	13.72	12.43 M
EV-VGCNN	P	0.959	0.46	0.82 M
VMV-GCN	P	0.975	0.33	0.84 M
Ours	P	0.985	**0.14**	**0.72 M**
Ours w/ CRD	P	0.983	**0.14**	**0.72 M**

Table 3: Comparison of models on the DVS128 dateset.

Methods	Type	N-Cal (mAP↑)
YOLE	F	0.398
Asynet	F	0.643
NvS-S	P	0.346
AEGNN	P	0.595
Ours	P	**0.657**
Ours w/ CRD	P	**0.711**

Table 4: Comparison of models $w.r.t$ the eleven-point mean average precision (mAP) on the object detection task.

Method	Type	#Params	GFLOPs	Time
EST	F	21.38 M	4.28	6.41 ms
M-LSTM	F	21.43 M	4.82	10.89 ms
MVF-Net	F	33.62 M	5.62	10.09 ms
EventNet	P	2.81 M	0.91	**3.35 ms**
PointNet++	P	1.76 M	4.03	103.85 ms
RG-CNNs	P	19.46 M	0.79	-
EV-VGCNN	P	0.84 M	0.70	7.12 ms
AEGNN	P	20.4 M	0.75	-
VMV-GCN	P	0.86 M	1.30	6.27 ms
Ours	P	**0.77 M**	**0.57**	3.84 ms

Table 5: Comparison of models on the model complexity (#Params) and the number of FLOPs.

the effectiveness of our proposed learning model. We attribute these improvements to the EDAL that can aggregate features for vertices considering all attributes dynamically, which allows us to efficiently and precisely extract the semantics of events.

Excitedly, the introduced CRD can improve the performance of the EDGCN by a large margin. It proves that our CRD can successfully utilize the pretrained weights of CNNs to ease the learning of our graph model and improve its representation ability, even with large image-to-graphs gaps. However, our model, largely improved by the CRD scheme, still lags behind some frame-based methods (*e.g.,* the MVF-Net) that are with pretraining. We attribute this to the much smaller discrepancy between image-frame than our frame-graph, allowing those frame-based methods to directly use the model weights well-trained with large-scale datasets. Moreover, Tab. 5 shows that our method holds large advantages in model complexity and computational cost over other approaches.

Action Recognition

In this section, we choose the action recognition task to validate the advantages of our model in encoding motions using the DVS128 (Amir et al. 2017) dataset which contains samples derived by different gestures. We follow (Bi et al. 2020) to sample all test data with 0.5s duration. The cross-entropy loss is utilized as $\mathcal{L}_{task}$ for the model's training. The I3D-R is chosen as the backbone of the frame-based teacher network when performing CRD for optimizing EDGCN and the performance of the teacher on DVS128 is 0.981.

Recognition Performance. Tab. 3 shows that our model has significant advantages over other point-based approaches $w.r.t$ recognition accuracy, model complexity, and computational cost. Despite using highly sparse input data, our model achieves accuracy comparable to the state-of-the-art frame-based method (TA-SNN). Notably, I3D-R shows weakness in this task compared to EDGCN even with pretraining and a much heavier model.

These findings suggest that EDGCN can accurately extract motion cues while preserving the sparsity of event data. This advantage can be attributed to two reasons. (i) Accurate neighborhood definition. (ii) Attentive aggregation which relies only on spatio-temporal relations for augmenting motion elements in feature representation. We also evaluate our work with aggregation considering joint features ($\mathcal{P}^{fuse}$) instead of only coordinates ($\mathcal{P}^{\mathcal{U}}$).

Object Detection

Event-based object detection is an emerging topic to simultaneously solve object localization and categorization. This task requires event-based models with powerful semantics and motion encoding capabilities. We conduct experimental comparisons for this task on the N-Cal dataset, which is a single object detection dataset containing 101 classes. The ResNet-34 is used as the backbone of the frame-based teacher branch for our CRD, and its performance on N-Cal is 0.76. Following the setup in (Schaefer, Gehrig, and Scaramuzza 2022), we use a collection of losses (a weighted sum of class, bounding box offset and shape as well as prediction confidence losses) as the $\mathcal{L}_{task}$ for training.

Detection Performance. We utilize the eleven-point mean average precision (mAP) to measure our models on the detection task. From results in Tab. 4 and 5, we conclude that our approach achieves large improvement on mAP over others with much fewer parameters and computational costs. More importantly, the proposed EDGCN trained solely exceeds other graph-based models such as NvS-S and AEGNN by a large margin, indicating the superiority of our EDGCN. Besides, CRD also largely boosts the detection performance in addition to object and action recognition tasks, suggesting the well generalizability of our proposed learning scheme.

Complexity Analysis

We compute the complexity and FLOPs of object classification models on the N-Cal dataset following (Bi et al. 2020; Deng et al. 2022). Results in Tab. 5 show that our approach holds extremely low model complexity and computational cost, indicating the high efficiency the learning system holds in extracting representative features from event

	AA_U	UPD	AA_{fuse}	$\mathcal{P}^U$	$\mathcal{P}^F$	$\mathcal{P}^{fuse}$	N-Cal	CIF10	DVS128
A	✓	✓		✓			0.793	0.695	0.954
B	✓	✓			✓		0.790	0.701	0.970
C	✓	✓				✓	**0.801**	**0.716**	**0.985**
D	✓					✓	0.783	0.694	0.972
E		✓	✓			✓	0.788	0.703	0.969

Table 6: The ablation study on the effects of different designs to model's performance. *AA_U: attentive aggregation using coordinate attributes. AA_{fuse}: attentive aggregation using the fused joint representation. UPD: coordinate attribute update module.*

EDGCN	$\mathcal{L}_{inf}$	$\mathcal{L}_{feat}$				N-Cal	CIF10
		A	B	C	D		
✓						0.801	0.716
✓	✓					0.818	0.724
✓	✓	✓				0.825	0.733
✓	✓		✓			0.830	0.749
✓	✓			✓		**0.835**	**0.752**
✓	✓				✓	0.814	0.721

Table 7: Effects of different designs to model's performance.

data. We measure inference time on the N-C using PyTorch on an Nvidia RTX 3090 and an Intel i7-13700. Our approach achieves leading performance with only 3.84 ms processing time per sample (equivalent to 260 Hz frame-rate), demonstrating its practical value in high-speed scenarios.

Ablation Study

In this section, we evaluate our method through various settings, where setup in Tab. 6 is for evaluating core modules of EDAL and Tab. 7 is for verify the contribution of each constraint in the proposed CRD. A visualization is also presented to show the benefits of CRD in improving the representation ability of EDGCN.

Effectiveness of Learning Modules in EDAL. We investigate the effectiveness of our neighborhood definition method through settings A, B, and C in Tab. 6, where A, B, and C represent defining neighbors using only coordinate attributes ($\mathcal{P}^U$), only semantic features ($\mathcal{P}^F$), and a joint representation ($\mathcal{P}^{fuse}$) of all attributes respectively. The results show that our neighborhood definition method consistently improves model performance across different datasets. In particular, compared to the method of finding neighbors using only coordinates as done in (Deng et al. 2022; Li et al. 2021; Schaefer, Gehrig, and Scaramuzza 2022), our method achieves significant improvement in action recognition tasks. This confirms our observation in Fig. 2 that it is difficult to find neighbors highly related to a vertex's motion and semantic information based solely on coordinates in scenarios where motion states are complex.

Additionally, we explore the impact of attentive aggregation mode on model performance (C & E). By enhancing the contribution of motion information during aggregation, our method can obtain more efficient graph representation. This validates the rationality of our design. Also, we can see that UPD further brings considerable improvement (C & D), indicating that the coordinate attribute strengthened by UPD can further facilitate the aggregation process.

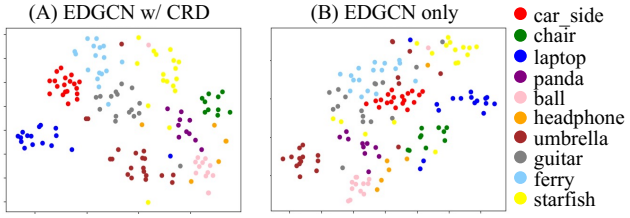


Figure 6: The t-SNE visualization on the test set of N-Cal.

Designs in the CRD. In Fig. 5, we introduce three intermediate feature-level knowledge transfer designs, A, B, and C, which all employ the contrastive loss to ensure consistency between teacher and student features. Variant B, however, differs from A by adding task-specific loss to both student and teacher branches. In contrast, C only applies the task-specific loss to the teacher branch. Moreover, D in Tab. 7 replaces the contrastive loss in A with hard constraints such as L1 loss (Romero et al. 2015) for feature distillation.

Tab. 7 shows that inference-level distillation outperforms the EDGCN trained solely. Furthermore, A, B, and C variants boost EDGCN by supervising intermediate features, with C being the best. This is due to $\mathcal{L}_{task}$ on intermediate features, which makes the teacher represent events more comprehensively and provide better guidance to the student. Surprisingly, B is worse than C with extra task loss for the student, potentially due to the limited power of intermediate features for reliable prediction. Furthermore, variant D degrades the entire learning framework, likely because of different learning logic of two models. This supports our choice of contrastive loss for frame-to-graph distillation, as different feature dimensions may have different semantics and hard constraints are not good for feature consistency.

Visualization for Feature Representation. We further illustrate t-SNE of features on the N-Cal dataset in Fig. 6. The EDGCN w/o CRD shows weakness in achieving explicit decision boundaries among some challenging classes such as car_side and ferry due to their similar appearances (Fig. 6.(B)). This limitation of EDGCN can be mitigated by CRD, where more discriminative features are obtained (Fig. 6.(A)), indicating the efficacy of CRD in boosting the representation ability of EDGCN.

Conclusion

We propose a novel event-based GCN (EDGCN) for defining each event-based vertex's neighborhood considering all its attributes and dynamically updates vertex attributes layer-by-layer. We also introduce a cross-representation distillation framework (CRD) for point-based methods that leverages large-scale prior from frame-based models to facilitate EDGCN training. Comprehensive experiments on various vision tasks validate the efficacy of our EDGCN and CRD. Since CRD has potential for migrating to other point-based methods, we argue that this learning strategy may open new research avenues for event-based model learning.

Acknowledgments

This work is jointly supported by National Key Research and Development Program of China (2022YFF0610000), the National Natural Science Foundation of China (62203024, 92167102, 61873220, 62102083, 62173286, 61875068, 62177018), the Natural Science Foundation of Jiangsu Province (BK20210222), the R&D Program of Beijing Municipal Education Commission (KM202310005027), the Research Grants Council of Hong Kong (CityU 11213420).

References

Amir, A.; Taba, B.; Berg, D.; Melano, T.; McKinstry, J.; Di Nolfo, C.; Nayak, T.; Andreopoulos, A.; Garreau, G.; Mendoza, M.; et al. 2017. A low power, fully event-based gesture recognition system. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 7243–7252.

Baldwin, R.; Liu, R.; Almatrafi, M. M.; Asari, V. K.; and Hirakawa, K. 2022. Time-Ordered Recent Event (TORE) Volumes for Event Cameras. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1–1.

Bi, Y.; Chadha, A.; Abbas, A.; Bourtsoulatze, E.; and Andreopoulos, Y. 2020. Graph-based Spatio-Temporal Feature Learning for Neuromorphic Vision Sensing. *IEEE Trans. Image Process.*, 1–1.

Cannici, M.; Ciccone, M.; Romanoni, A.; and Matteucci, M. 2019. Asynchronous convolutional networks for object detection in neuromorphic cameras. In *IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, 0–0.

Chen, C.-F. R.; Panda, R.; Ramakrishnan, K.; Feris, R.; Cohn, J.; Oliva, A.; and Fan, Q. 2021. Deep analysis of cnn-based spatio-temporal representations for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6165–6175.

Chen, J.; Meng, J.; Wang, X.; and Yuan, J. 2020a. Dynamic Graph CNN for Event-Camera Based Gesture Recognition. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–5.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020b. A simple framework for contrastive learning of visual representations. In *Int. Conf. Mach. Learn*, 1597–1607. PMLR.

Deng, Y.; Chen, H.; Chen, H.; and Li, Y. 2021. Learning from Images: A Distillation Learning Framework for Event Cameras. *IEEE Trans. Image Process.*, 1–1.

Deng, Y.; Chen, H.; and Li, Y. 2021. MVF-Net: A Multi-view Fusion Network for Event-based Object Classification. *IEEE Trans. Circuits Syst. Video Technol.*, 1–1.

Deng, Y.; Chen, H.; Liu, H.; and Li, Y. 2022. A Voxel Graph CNN for Object Classification With Event Cameras. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 1172–1181.

Deng, Y.; Li, Y.; and Chen, H. 2020. AMAE: Adaptive Motion-Agnostic Encoder for Event-Based Object Classification. *IEEE Robot. Autom. Lett.*, 5(3): 4596–4603.

Gehrig, D.; Gehrig, M.; Hidalgo-Carrió, J.; and Scaramuzza, D. 2020. Video to Events: Recycling Video Datasets for Event Cameras. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Gehrig, D.; Loquercio, A.; Derpanis, K. G.; and Scaramuzza, D. 2019. End-to-end learning of representations for asynchronous event-based data. In *IEEE/CVF Int. Conf. Comput. Vis.*, 5633–5643.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning and Representation Learning Workshop*.

Hu, L.; Zhao, R.; Ding, Z.; Ma, L.; Shi, B.; Xiong, R.; and Huang, T. 2022. Optical Flow Estimation for Spiking Camera. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 17844–17853.

Hu, Y.; Delbruck, T.; and Liu, S.-C. 2020. Learning to Exploit Multiple Vision Modalities by Using Grafted Networks. In *Eur. Conf. Comput. Vis.*, 85–101.

Huang, S.; Xie, Y.; Zhu, S.-C.; and Zhu, Y. 2021. Spatio-temporal self-supervised representation learning for 3d point clouds. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 6535–6545.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Li, H.; Liu, H.; Ji, X.; Li, G.; and Shi, L. 2017. Cifar10-dvs: An event-stream dataset for object classification. *Front. Neurosci.*, 11: 309.

Li, Y.; Zhou, H.; Yang, B.; Zhang, Y.; Cui, Z.; Bao, H.; and Zhang, G. 2021. Graph-based Asynchronous Event Processing for Rapid Object Recognition. In *IEEE/CVF Int. Conf. Comput. Vis.*, 914–923.

Li, Z.; Asif, M. S.; and Ma, Z. 2022. Event Transformer. *arXiv preprint arXiv:2204.05172*.

Messikommer, N.; Gehrig, D.; Gehrig, M.; and Scaramuzza, D. 2022. Bridging the Gap between Events and Frames through Unsupervised Domain Adaptation. *IEEE Robot. Autom. Lett.*, 7(2): 3515–3522.

Messikommer, N.; Gehrig, D.; Loquercio, A.; and Scaramuzza, D. 2020. Event-based asynchronous sparse convolutional networks. In *Eur. Conf. Comput. Vis.*, 415–431. Springer.

Mitrokhin, A.; Hua, Z.; Fermuller, C.; and Aloimonos, Y. 2020. Learning visual motion segmentation using event surfaces. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 14414–14423.

Orchard, G.; Jayawant, A.; Cohen, G. K.; and Thakor, N. 2015. Converting static image datasets to spiking neuromorphic datasets using saccades. *Front. Neurosci.*, 9: 437.

Orchard, G.; Meyer, C.; Etienne-Cummings, R.; Posch, C.; Thakor, N.; and Benosman, R. 2015. HFirst: A Temporal Approach to Object Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(10): 2028–2040.

Rebecq, H.; Gehrig, D.; and Scaramuzza, D. 2018. ESIM: an open event camera simulator. In *Conf. on Robot Learn*, 969–982. PMLR.

Rebecq, H.; Ranftl, R.; Koltun, V.; and Scaramuzza, D. 2019. High Speed and High Dynamic Range Video with an Event Camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1–1.

Rebecq, H.; Ranftl, R.; Koltun, V.; and Scaramuzza, D. 2019. High Speed and High Dynamic Range Video with an Event Camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1–1.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 779–788.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2015. FitNets: Hints for Thin Deep Nets. In *Int. Conf. Learn. Represent.*

Schaefer, S.; Gehrig, D.; and Scaramuzza, D. 2022. AEGNN: Asynchronous Event-based Graph Neural Networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Sekikawa, Y.; Hara, K.; and Saito, H. 2019. EventNet: Asynchronous Recursive Event Processing. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Sironi, A.; Brambilla, M.; Bourdis, N.; Lagorce, X.; and Benosman, R. 2018. HATS: Histograms of averaged time surfaces for robust event-based object classification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 1731–1740.

Sun, Z.; Messikommer, N.; Gehrig, D.; and Scaramuzza, D. 2022. ESS: Learning Event-based Semantic Segmentation from Still Images. In *Eur. Conf. Comput. Vis.*, 341–357. Springer.

Tulyakov, S.; Bochicchio, A.; Gehrig, D.; Georgoulis, S.; Li, Y.; and Scaramuzza, D. 2022. Time Lens++: Event-based Frame Interpolation with Parametric Non-linear Flow and Multi-scale Fusion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 17755–17764.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wang, L.; Huang, Y.; Hou, Y.; Zhang, S.; and Shan, J. 2019a. Graph Attention Convolution for Point Cloud Semantic Segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10288–10297.

Wang, Q.; Zhang, Y.; Yuan, J.; and Lu, Y. 2019b. Space-Time Event Clouds for Gesture Recognition: From RGB Cameras to Event Cameras. In *IEEE Winter Conf. on Appl. of Comput. Vis.*, 1826–1835.

Wu, Z.; Zhang, H.; Lin, Y.; Li, G.; Wang, M.; and Tang, Y. 2022. LIAF-Net: Leaky Integrate and Analog Fire Network for Lightweight and Efficient Spatiotemporal Information Processing. *IEEE Transactions on Neural Networks and Learning Systems*, 33(11): 6249–6262.

Xie, B.; Deng, Y.; Shao, Z.; Liu, H.; and Li, Y. 2022. VMV-GCN: Volumetric Multi-View Based Graph CNN for Event Stream Classification. *IEEE Robot. Autom. Lett.*, 7(2): 1976–1983.

Yao, M.; Gao, H.; Zhao, G.; Wang, D.; Lin, Y.; Yang, Z.; and Li, G. 2021. Temporal-wise attention spiking neural networks for event streams classification. In *Int. Conf. Comput. Vis.*, 10221–10230.

Zhu, L.; Wang, X.; Chang, Y.; Li, J.; Huang, T.; and Tian, Y. 2022. Event-Based Video Reconstruction via Potential-Assisted Spiking Neural Network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 3594–3604.