MATRIX-HAR: Lightweight Temporal Motion-guided Feature Network with Adaptive Temporal Resolution Based Transfer Learning for Resource-Constrained Event Camera-based Human Action Recognition

Muhammad Hamza Zafar[a], Syed Kumayl Raza Moosavi[a] and Filippo Sanfilippo[a,*]

[a]Department of Engineering Sciences, University of Agder, Grimstad, 4879, Norway

ARTICLE INFO

Keywords:
Event Camera
Human-Robot Collaboration
Transfer Learning
Human Action Recognition
Neuromorphic Vision

ABSTRACT

Event-based cameras have emerged as a promising alternative to traditional sensors for human action recognition due to their high temporal resolution, low latency, and robustness to lighting variations. However, existing methods struggle to effectively capture the complex spatio-temporal dynamics inherent in event streams, particularly under challenging environmental conditions. In this paper, we propose MATRIX-HAR, a novel deep learning architecture specifically designed for event-based human action recognition that incorporates three key innovations: an Adaptive Temporal Resolution (ATR) module that dynamically adjusts temporal processing granularity based on input characteristics, a Motion Guided Feature Module (MGFM) that leverages intrinsic motion information encoded in event streams, and a comprehensive hierarchical transfer learning framework that enables effective knowledge transfer from simple to challenging environmental settings. We systematically evaluate four distinct transfer learning strategies, ranging from early feature transfer to comprehensive multi-level transfer, demonstrating their effectiveness in adapting models trained on simple scenarios to complex real-world conditions. To ensure practical deployment on resource-constrained edge devices, we implement post-training quantization that maintains performance while achieving 4× compression ratios. Extensive experiments on four benchmark datasets (HARDVS, PAF, THU-E-ACT-50, and THU-E-ACT-50 CHL) demonstrate that MATRIX-HAR achieves state-of-the-art performance, surpassing existing methods by significant margins: 0.71% on HARDVS (97.82%), 0.74% on PAF (97.23%), 3.03% on THU-E-ACT-50 (95.73%), and 7.3% on THU-E-ACT-50 CHL (65.8%). Ablation studies confirm the individual contributions of each proposed module, while transfer learning analysis reveals that comprehensive feature transfer (Strategy 4) provides the most robust adaptation with 22.3% relative performance improvement over baseline approaches. The proposed framework demonstrates superior learning dynamics, achieving faster convergence and enhanced stability compared to training from scratch, making it particularly suitable for real-world deployment scenarios where environmental conditions are variable and computational resources are limited.

1. Introduction

Industrial robotics has witnessed a significant transformation, from early isolated, task-specific machines to modern systems capable of real-time collaboration with human workers. At the heart of this evolution lies Human-Robot Collaboration (HRC), where robots are expected to operate safely and efficiently alongside humans in shared workspaces [1]. For such collaboration to be successful, robots must accurately perceive and interpret human actions, gestures, and intent in dynamic, real-world environments.

Traditional HRC systems predominantly rely on vision-based sensors, particularly RGB cameras, to monitor and analyze human activities [2]. While effective in controlled environments, these cameras face major limitations in variable lighting conditions, high-speed motion, and occlusion-prone settings [3]. Their frame-based nature also results in temporal sparsity and data redundancy, making them less suited for applications demanding precise, low-latency action recognition.

Neuromorphic event cameras have emerged as a powerful alternative to address these challenges. Unlike conventional cameras that capture entire frames at fixed intervals, event cameras asynchronously record changes in pixel intensity, offering microsecond-level temporal resolution and high dynamic range [4]. These properties make them ideal for capturing fast, fine-grained motion patterns under diverse environmental conditions, including low light and high-speed scenarios.

Accurately analyzing the sparse and inconsistent data from event cameras is difficult, especially for recognizing human actions (HAR) during real-world human-robot collaboration. Existing deep learning (DL) models often don't fully use the detailed timing information within event streams, which limits their effectiveness in complicated situations.

*Corresponding author

muhammad.h.zafar@uia.no (M.H. Zafar); syed.k.moosavi@uia.no (S.K.R. Moosavi); filippo.sanfilippo@uia.no (F. Sanfilippo)

ORCID(s):

2. Related Work

2.1. Event Data Processing Approaches

Event cameras represent a paradigm shift in visual sensing technology, but their unique asynchronous data format presents significant implementation challenges. Researchers have developed various approaches to handle this distinctive output format. Event aggregation techniques have been a primary focus, transforming asynchronous signals into synchronized visual representations, as seen in the work of Ghosh et al. [5] and Innocenti et al. [6]. While early research by Baby et al. [7] and Clady et al. [8] utilized handcrafted features, the field has increasingly embraced deep learning approaches. Innocenti et al. [6] proposed an innovative method converting binary representations into decimal-based frames by assessing event presence within defined time intervals.

2.2. Conventional Deep Learning Approaches

The development of biologically-inspired visual architectures marked a significant advancement in analyzing human movements. Layher et al. [9] introduced a CNN-based model demonstrating exceptional energy efficiency—processing approximately 1,000 frames per second while consuming only 70 mW. This work complemented Baby et al.'s [7] approach combining dynamic vision sensor data with Motion Maps and Motion Boundary Histograms, achieving results comparable to traditional video-based methods. The integration of neuromorphic vision sensors with sophisticated deep learning frameworks has substantially enhanced action recognition capabilities. Pradhan et al. [10] adapted event memory surfaces to make sparse data compatible with CNNs, while Ghosh et al. [5] addressed inherent challenges of noise and sparsity through spatiotemporal filtering techniques. Despite these advances, Pansuriya et al. [11] and Huang et al. [12] note that challenges persist in effectively encoding spatiotemporal data for complex recognition tasks in diverse real-world scenarios. Novel data representations and architectural approaches have emerged to better utilize event data characteristics. Liu et al. [13] proposed the aligned compressed event tensor (ACE) and branched event network (BET) framework, reporting 98.88% accuracy on the DVS128 dataset. For privacy-preserving applications, Chen et al. [14] and Prasad et al. [15] explored temporal localization frameworks with spatiotemporal attention mechanisms for fall detection. In egocentric recognition applications, Moreno et al. [16] performed comparative analysis of event-based and traditional visual data, revealing that event frames provide distinctive information enhancing recognition performance. Plizzari et al. [17] introduced the valuable N-EPIC-Kitchens dataset, while Ren et al. [18] developed the innovative TTPOINT framework. Cao et al. [19] created the EventCrab framework that effectively integrates frame-specific and point-specific network strengths. Wang et al.'s [20] HARDVS dataset has provided valuable benchmarking resources, while Gao et al. [21] demonstrated the effectiveness of their HyperMV framework in multi-view recognition scenarios. Plou et al. [22] showcased potential medical applications with the EventSleep dataset, and Cultrera et al. [23] applied event camera technology to facial micro-expression recognition using a novel spatio-temporal Vision Transformer model trained on the FACEMORPHIC dataset.

2.3. Spiking Neural Network Approaches

Spiking neural networks (SNNs) have emerged as a promising alternative for processing event-based data. Liu et al. [24] explored SNNs as a potential solution for processing motion features in event-based cameras, offering a hierarchical approach that processes motion information from local to global levels. Yang et al. [25] leveraged spiking neural networks to address the noise and sparsity inherent in event-based data. For energy-efficient human activity recognition tasks, Bian et al. [26] investigated the application of SNNs. Ren et al. [27] introduced the SpikePoint architecture, representing another significant advancement. Lin et al. [28] developed the Spike-HAR and Spike-HAR++ models, introducing spiking transformers specifically designed for event-based human activity recognition. The field continues evolving toward optimized algorithms that better handle the sparse, asynchronous nature of event data while extending these systems to more complex real-world applications.

2.4. Contributions

Our proposed methodology, as shown in Fig. 1 follows a comprehensive pipeline designed to address the challenges of event-based human action recognition in varying environmental complexities. The framework begins with event stream acquisition from neuromorphic cameras, which are subsequently converted into frame representations through an Events-to-Frame Converter. The core training paradigm involves a two-stage approach: first, the MATRIX-HAR model is trained on the THU-E-ACT-50 dataset (simple environmental settings) to learn fundamental spatio-temporal patterns and action representations. Subsequently, we employ hierarchical transfer learning strategies to adapt the learned knowledge to the challenging THU-E-ACT-50 CHL dataset, where different architectural layers are selectively transferred based on four distinct strategies. Following the transfer learning phase, both models undergo quantization to optimize computational efficiency while preserving performance. The final deployment phase involves comprehensive evaluation of human action recognition capabilities, generating classification metrics for both the original model and transfer learning variants. This systematic approach enables effective knowledge transfer from simple to complex environmental conditions while maintaining computational efficiency through model compression, making the framework suitable for real-world deployment on resource-constrained edge devices.

The contributions of the proposed work are presented below:

- **Adaptive Temporal Resolution (ATR) Module:** We introduce a novel adaptive temporal resolution mechanism that dynamically adjusts the temporal processing

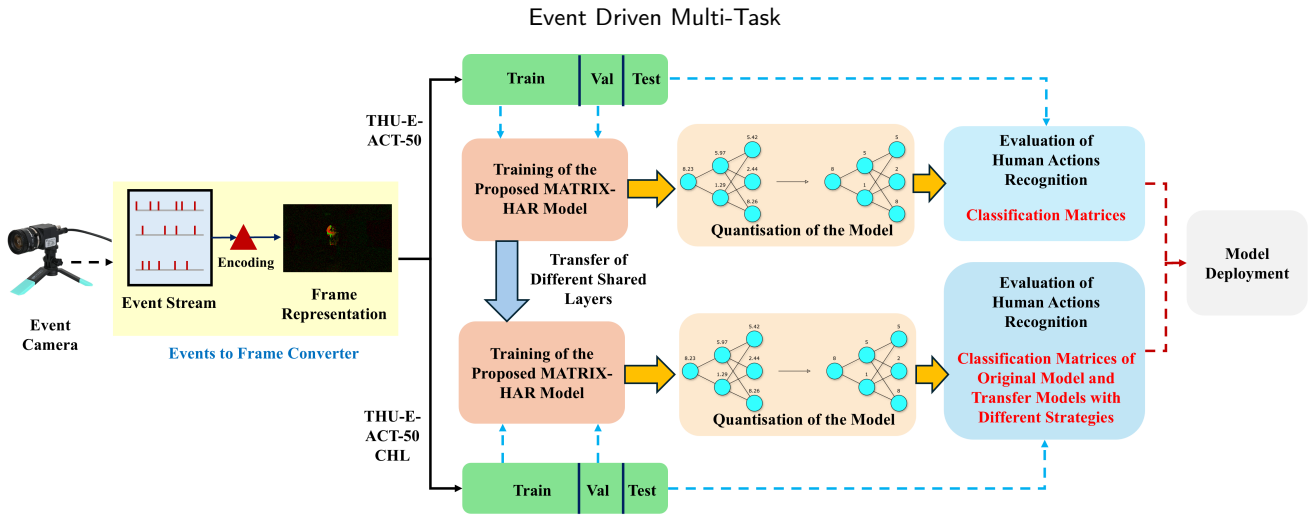

Figure 1: Detailed Description of Different Steps of the Proposed Work

granularity based on input characteristics, enabling the model to efficiently handle varying action durations and temporal complexities inherent in event-based data. This module significantly improves recognition accuracy by optimizing temporal feature extraction for different action patterns.

- **Motion Guided Feature Module (MGFM):** We propose a specialized motion-guided feature extraction module that leverages the intrinsic motion information encoded in event streams to enhance discriminative feature learning. The MGFM effectively captures motion-specific patterns that are crucial for event-based action recognition, leading to improved performance across all benchmark datasets.

- **Hierarchical Transfer Learning Strategies for Event-based Action Recognition:** We develop and systematically evaluate four novel transfer learning strategies specifically designed for event-based human action recognition, enabling effective knowledge transfer from simple to challenging environmental settings. Our comprehensive transfer learning approach demonstrates superior adaptability and robustness, particularly Strategy 4 (comprehensive feature transfer), which achieves significant performance improvements while maintaining efficiency through quantization-aware deployment.

3. Dataset Description

In this study 4 benchmark datasets are used for the evaluation of the proposed MATRIX-HAR model but for transfer learning approaches only THU-E-ACT-50 and THU-E-ACT-50 CHL is used.

1. **PAF [29]**: The action recognition dataset comprises event-based recordings captured using the DAVIS346 neuromorphic vision sensor, featuring 15 subjects performing 10 distinct actions—arm-crossing, getting-up, kicking, picking-up, jumping, sitting-down, throwing, turning-around, walking, and waving—in an empty office environment. Each subject performs each action three times, with recordings lasting approximately five seconds each, captured from three different camera positions and distances to ensure variability. The dataset includes a total of 450 recordings at a resolution of 346 × 260 pixels, recorded as asynchronous event streams in .aedat format.
2. **HARDVS [20]**: The HARDVS dataset is a large-scale benchmark for human activity recognition using event-based neuromorphic vision sensors, captured with a DAVIS346 camera. It comprises over 100,000 event sequences across 300 diverse human activity categories reflecting daily life actions such as drinking, riding a bike, and washing hands. Each video clip lasts approximately 5 to 10 seconds and is recorded under varied conditions including multiple viewpoints (front, side, top-down, etc.), lighting scenarios (strong, medium, low illumination), motion speeds (slow to fast), dynamic backgrounds, and occlusions, thereby simulating real-world challenges.
3. **THU-E-ACT-50 [30]**: The THU-E-ACT-50 dataset is a large-scale event-based action recognition benchmark comprising 10,500 video sequences captured using high-resolution CeleX-V event cameras at 1280 × 800 pixels. It features 105 subjects performing 50 diverse action categories, including whole-body movements, healthcare-related actions, detailed hand gestures, and human-object interactions. The recordings were captured from two synchronized frontal views in a spacious indoor environment, offering multi-view perspectives and a wide demographic range across age and gender. This dataset significantly expands the scale and variety of event-based action recognition data, supporting research in practical applications

such as healthcare monitoring and social interaction analysis.

4. **THU-E-ACT-50-CHL [30]**: The THU-E-ACT-50-CHL dataset complements the standard benchmark by providing 2,330 sequences recorded under challenging environmental conditions using DAVIS346 event cameras at a resolution of 346 × 260 pixels. Captured from multiple perspectives including front, left, right, and occasionally back views, this dataset reflects real-world difficulties such as low illumination, dynamic backgrounds, occlusions, and noisy event streams. The participants, 18 young adults, performed the same 50 action categories as in THU-E-ACT-50, with an emphasis on robustness testing for action recognition methods. The THU-E-ACT-50-CHL dataset serves as a rigorous benchmark to evaluate algorithm performance under adverse and practical deployment scenarios.

4. Proposed Model: MATRIX-HAR

The complete MATRIX-HAR model architecture integrates the Motion Guided Feature Modulation (MGFM) and ATR modules within a 3D CNN backbone to effectively capture spatial and temporal dynamics across multiple scales.

Let $\mathcal{X}_{\text{input}} \in \mathbb{R}^{B\times T\times 3\times H\times W}$ be the input video sequence, where B is the batch size, T is the sequence length, and H, W are the spatial dimensions. The input is first permuted to the format expected by 3D convolutions:

$$\mathcal{X}_0 = \text{Permute}(\mathcal{X}_{\text{input}}) \in \mathbb{R}^{B\times 3\times T\times H\times W} \tag{1}$$

4.1. Feature Extraction and Progressive Processing

The model processes features through a sequence of operations, incorporating the specialized MGFM and ATR modules at strategic positions:

$$\mathcal{X}_1 = \text{ReLU}(\text{GN}(\mathcal{W}_1 * \mathcal{X}_0 + \mathbf{b}_1)) \in \mathbb{R}^{B\times C/2\times T\times H\times W} \tag{2}$$

where $\mathcal{W}_1 \in \mathbb{R}^{C/2\times 3\times 3\times 3\times 3}$ and $\mathbf{b}_1 \in \mathbb{R}^{C/2}$ are the weights and biases of the first 3D convolution, and $\text{GN}(\cdot)$ denotes Group Normalization.

4.1.1. Early Motion-Guided Feature Modulation

The first MGFM module is applied to enhance motion features early in the network:

$$\mathcal{X}_2 = \text{MGFM}_{\text{early}}(\mathcal{X}_1) \in \mathbb{R}^{B\times C/2\times T\times H\times W} \tag{3}$$

where $\text{MGFM}_{\text{early}}(\cdot)$ is the first Motion Guided Feature Modulation module as defined in Section 1.

4.1.2. Second Convolution and Pooling

$$\mathcal{X}_3 = \text{ReLU}(\text{GN}(\mathcal{W}_2 * \mathcal{X}_2 + \mathbf{b}_2)) \tag{4}$$

$$\mathcal{X}_4 = \text{MaxPool3D}(\mathcal{X}_3 \tag{5}$$

where $\mathcal{W}_2 \in \mathbb{R}^{C\times C/2\times 3\times 3\times 3}$ and $\mathbf{b}_2 \in \mathbb{R}^{C}$ are the weights and biases of the second 3D convolution.

4.1.3. Adaptive Temporal Resolution

The ATR module is applied after the first pooling operation to process features at multiple temporal scales:

$$\mathcal{X}_5 = \text{ATR}(\mathcal{X}_4) \in \mathbb{R}^{B\times C\times T/2\times H/2\times W/2} \tag{6}$$

where $\text{ATR}(\cdot)$ is the Adaptive Temporal Resolution module as defined in Section 2.

4.1.4. Mid-Level Motion-Guided Feature Modulation

The second MGFM module leverages the enhanced multi-scale temporal features:

$$\mathcal{X}_6 = \text{MGFM}_{\text{mid}}(\mathcal{X}_5) \in \mathbb{R}^{B\times C\times T/2\times H/2\times W/2} \tag{7}$$

where $\text{MGFM}_{\text{mid}}(\cdot)$ is the second Motion Guided Feature Modulation module.

4.1.5. Temporal Attention

A temporal attention mechanism is applied to focus on the most relevant temporal segments:

$$\mathcal{A} = \sigma(\mathcal{W}_a * \mathcal{X}_6 + \mathbf{b}_a) \in \mathbb{R}^{B\times 1\times T/2\times H/2\times W/2} \tag{8}$$

$$\mathcal{X}_7 = \mathcal{X}_6 \odot \mathcal{A} \in \mathbb{R}^{B\times C\times T/2\times H/2\times W/2} \tag{9}$$

where $\mathcal{W}_a \in \mathbb{R}^{1\times C\times 1\times 1\times 1}$ and $\mathbf{b}_a \in \mathbb{R}^{1}$ are the weights and biases of the attention convolution, and $\sigma(\cdot)$ is the sigmoid activation function.

4.1.6. Third Convolution and Pooling

$$\mathcal{X}_8 = \text{ReLU}(\text{GN}(\mathcal{W}_3 * \mathcal{X}_7 + \mathbf{b}_3)) \tag{10}$$

$$\mathcal{X}_9 = \text{MaxPool3D}(\mathcal{X}_8 \tag{11}$$

where $\mathcal{W}_3 \in \mathbb{R}^{C\times C\times 3\times 3\times 3}$ and $\mathbf{b}_3 \in \mathbb{R}^{C}$ are the weights and biases of the third 3D convolution.

4.1.7. Late Motion-Guided Feature Modulation

The final MGFM module refines the high-level spatio-temporal features:

$$\mathcal{X}_{10} = \text{MGFM}_{\text{late}}(\mathcal{X}_9) \in \mathbb{R}^{B\times C\times T/4\times H/4\times W/4} \tag{12}$$

where $\text{MGFM}_{\text{late}}(\cdot)$ is the third Motion Guided Feature Modulation module.

4.1.8. Global Average Pooling

Global average pooling is applied to aggregate features across all dimensions except the batch and channel dimensions:

$$\mathcal{X}_{11} = \text{AdaptiveAvgPool3D}(\mathcal{X}_{10}, (1,1,1)) \in \mathbb{R}^{B\times C\times 1\times 1\times 1} \tag{13}$$

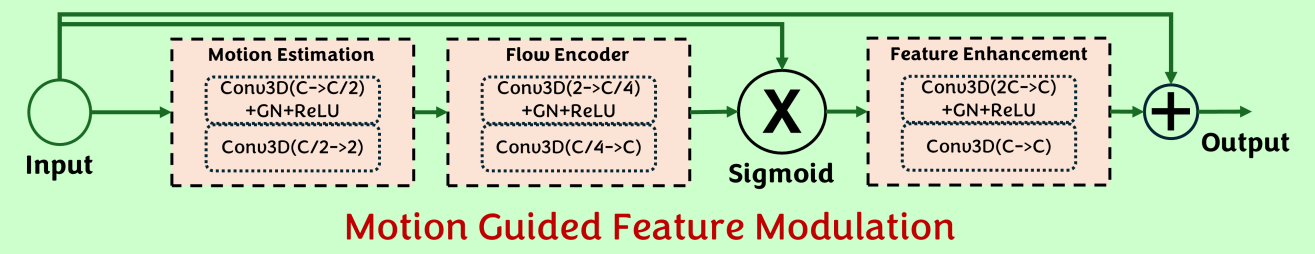

Figure 2: Detailed Internal Structure of Motion Guided Feature Modulation Module

4.1.9. Flattening and Classification

The pooled features are flattened and passed through fully connected layers for classification:

$$\mathcal{X}_{12} = \text{Flatten}(\mathcal{X}_{11}) \in \mathbb{R}^{B\times C} \tag{14}$$

$$\mathcal{X}_{13} = \text{ReLU}(\mathcal{W}_f^1 \mathcal{X}_{12} + \mathbf{b}_f^1) \in \mathbb{R}^{B\times 512} \tag{15}$$

$$\mathcal{X}_{14} = \text{Dropout}(\mathcal{X}_{13}, p = \text{dropout_rate}) \in \mathbb{R}^{B\times 512} \tag{16}$$

$$\mathcal{Y} = \mathcal{W}_f^2 \mathcal{X}_{14} + \mathbf{b}_f^2 \in \mathbb{R}^{B\times N_{\text{classes}}} \tag{17}$$

where $\mathcal{W}_f^1 \in \mathbb{R}^{512\times C}$ and $\mathbf{b}_f^1 \in \mathbb{R}^{512}$ are the weights and biases of the first fully connected layer, $\mathcal{W}_f^2 \in \mathbb{R}^{N_{\text{classes}}\times 512}$ and $\mathbf{b}_f^2 \in \mathbb{R}^{N_{\text{classes}}}$ are the weights and biases of the second fully connected layer, Dropout$(\cdot, p)$ applies dropout with probability p, and N_{classes} is the number of classes for the classification task

The model is trained using cross-entropy loss:

$$\mathcal{L}_{\text{CE}}(\mathcal{Y}, \mathbf{y}) = -\frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{N_{\text{classes}}} \mathbf{y}_{i,j} \log\left(\frac{e^{\mathcal{Y}_{i,j}}}{\sum_{k=1}^{N_{\text{classes}}} e^{\mathcal{Y}_{i,k}}}\right) \tag{18}$$

where $\mathbf{y} \in \mathbb{R}^{B\times N_{\text{classes}}}$ is the one-hot encoded ground truth.

The optimization is performed using the Adam optimizer with weight decay:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} - \lambda \cdot \theta_t \tag{19}$$

where θ_t represents the model parameters at step t, η is the learning rate, $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected first and second moment estimates, λ is the weight decay coefficient, and ϵ is a small constant for numerical stability

4.2. Motion Guided Feature Modulation (MGFM)

The Motion Guided Feature Modulation (MGFM) module enhances feature representations by extracting motion information and using it to modulate spatial features. This enables the network to dynamically focus on motion-relevant regions. The detailed architecture of MGFM module is shown in Fig. 2.

Let $\mathcal{X} \in \mathbb{R}^{B\times C\times T\times H\times W}$ be the input tensor to the MGFM module, where B is the batch size, C is the number of channels, T is the temporal dimension, and H, W are the spatial dimensions.

4.2.1. Motion Estimation Branch

The motion estimation branch extracts motion features through a series of 3D convolutions:

$$\mathcal{M}_1 = \text{ReLU}(\text{GN}(\mathcal{W}_m^1 * \mathcal{X} + \mathbf{b}_m^1)) \tag{20}$$

$$\mathcal{M}_2 = \mathcal{W}_m^2 * \mathcal{M}_1 + \mathbf{b}_m^2 \tag{21}$$

where $\mathcal{W}_m^1 \in \mathbb{R}^{C/2\times C\times 3\times 3\times 3}$, $\mathbf{b}_m^1 \in \mathbb{R}^{C/2}$ are the weights and biases of the first 3D convolution, $\mathcal{W}_m^2 \in \mathbb{R}^{2\times C/2\times 3\times 3\times 3}$, $\mathbf{b}_m^2 \in \mathbb{R}^2$ are the weights and biases of the second 3D convolution, and GN$(\cdot)$ denotes Group Normalization with number of groups $\min(32, C/2)$ and $\mathcal{M}_2 \in \mathbb{R}^{B\times 2\times T\times H\times W}$ represents the estimated motion flow fields

4.2.2. Flow Encoding

The estimated motion flow is encoded into feature modulation weights:

$$\mathcal{F}_1 = \text{ReLU}(\text{GN}(\mathcal{W}_f^1 * \mathcal{M}_2 + \mathbf{b}_f^1)) \tag{22}$$

$$\mathcal{F}_2 = \mathcal{W}_f^2 * \mathcal{F}_1 + \mathbf{b}_f^2 \tag{23}$$

where $\mathcal{W}_f^1 \in \mathbb{R}^{C/4\times 2\times 3\times 3\times 3}$ and $\mathbf{b}_f^1 \in \mathbb{R}^{C/4}$ are the weights and biases of the first flow encoding convolution, $\mathcal{W}_f^2 \in \mathbb{R}^{C\times C/4\times 1\times 1\times 1}$ and $\mathbf{b}_f^2 \in \mathbb{R}^C$ are the weights and biases of the second flow encoding convolution, GN$(\cdot)$ denotes Group Normalization with number of groups $\min(8, C/4)$, and $\mathcal{F}_2 \in \mathbb{R}^{B\times C\times T\times H\times W}$ represents the motion-based modulation weights

4.2.3. Feature Modulation

The input features are modulated using the encoded flow information via element-wise multiplication with a sigmoid-activated version of the flow features:

$$\mathcal{X}_{\text{mod}} = \mathcal{X} \odot \sigma(\mathcal{F}_2) \tag{24}$$

where $\sigma(\cdot)$ represents the sigmoid activation function: $\sigma(x) = \frac{1}{1+e^{-x}}$, $\odot$ denotes element-wise multiplication (Hadamard product), and $\mathcal{X}_{\text{mod}} \in \mathbb{R}^{B\times C\times T\times H\times W}$ represents the modulated features

4.2.4. Feature Enhancement

The original and modulated features are concatenated and enhanced through additional convolutions:

$$\mathcal{X}_{\text{cat}} = [\mathcal{X}, \mathcal{X}_{\text{mod}}] \tag{25}$$

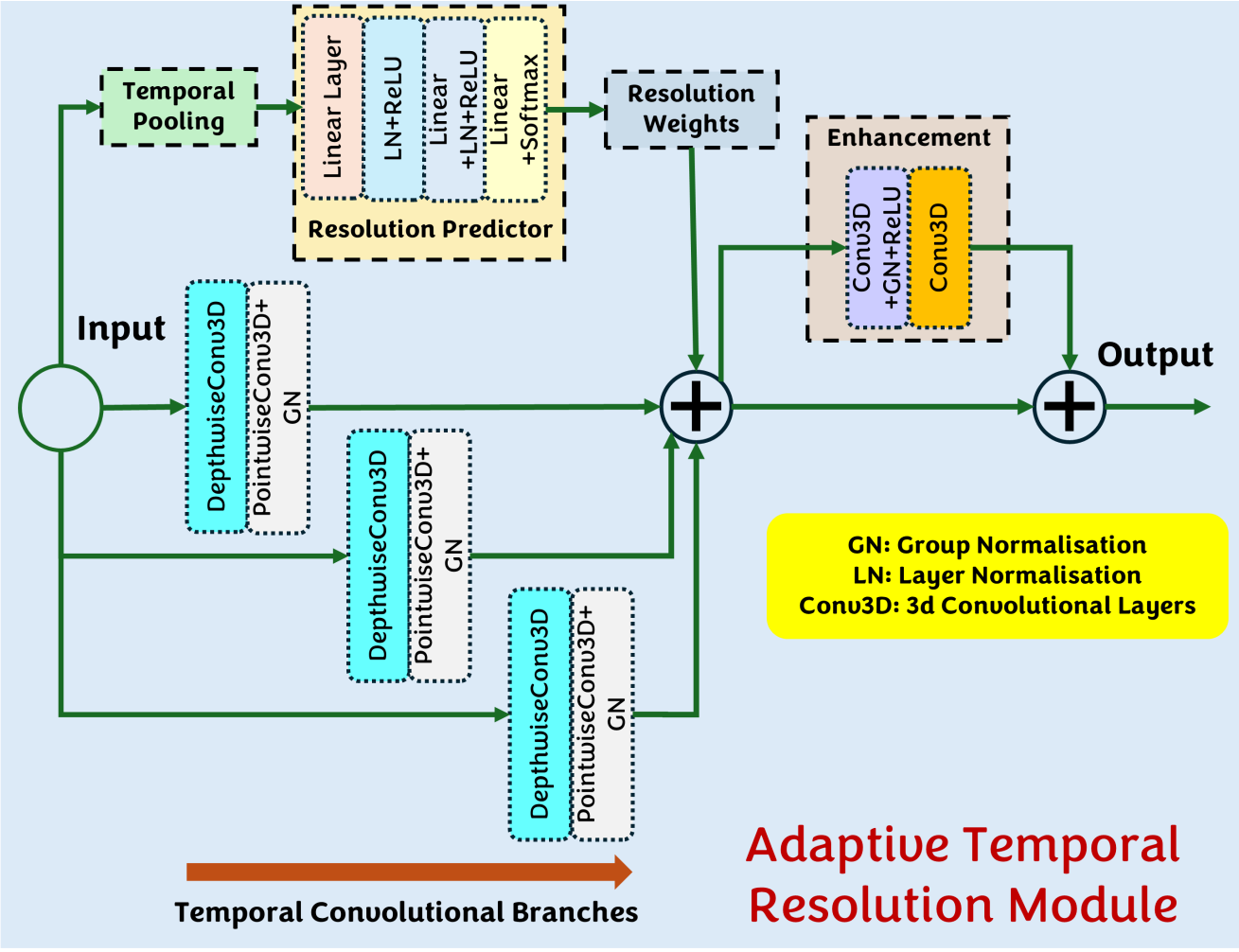

Figure 3: Detailed Internal Structure of Adaptive Temporal Resolution

$$\mathcal{E}_1 = \text{ReLU}(\text{GN}(\mathcal{W}_e^1 * \mathcal{X}_{\text{cat}} + \mathbf{b}_e^1)) \tag{26}$$

$$\mathcal{E}_2 = \mathcal{W}_e^2 * \mathcal{E}_1 + \mathbf{b}_e^2 \tag{27}$$

where $[\cdot,\cdot]$ denotes channel-wise concatenation, resulting in $\mathcal{X}_{\text{cat}} \in \mathbb{R}^{B\times 2C\times T\times H\times W}$, $\mathcal{W}_e^1 \in \mathbb{R}^{C\times 2C\times 1\times 1\times 1}$ and $\mathbf{b}_e^1 \in \mathbb{R}^C$ are the weights and biases of the first enhancement convolution, $\mathcal{W}_e^2 \in \mathbb{R}^{C\times C\times 1\times 1\times 1}$ and $\mathbf{b}_e^2 \in \mathbb{R}^C$ are the weights and biases of the second enhancement convolution, and $\text{GN}(\cdot)$ denotes Group Normalization with number of groups $\min(32, C)$

4.2.5. *Residual Connection*

Finally, a residual connection is added to preserve the original information:

$$\mathcal{Y} = \mathcal{E}_2 + \mathcal{X} \tag{28}$$

where $\mathcal{Y} \in \mathbb{R}^{B\times C\times T\times H\times W}$ is the output of the MGFM module.

4.3. Adaptive Temporal Resolution (ATR)

The Adaptive Temporal Resolution (ATR) module processes input features at multiple temporal scales concurrently and dynamically weights their contributions based on the sequence content. The detailed architecture of ATR module is shown in Fig. 3.

Let $\mathcal{X} \in \mathbb{R}^{B\times C\times T\times H\times W}$ be the input tensor to the ATR module.

4.3.1. *Temporal Pooling*

First, the input is temporally pooled to extract sequence-level information:

$$\mathcal{P} = \mathcal{F}_{\text{pool}}(\mathcal{X}) \tag{29}$$

where $\mathcal{F}_{\text{pool}}$ represents the adaptive average pooling operation that maintains the temporal dimension but pools spatial dimensions to 1×1, and $\mathcal{P} \in \mathbb{R}^{B\times C\times T\times 1\times 1}$ is the temporally pooled representation

4.3.2. *Resolution Predictor Network*

The pooled representation is then flattened and passed through a resolution predictor network to generate weights for each temporal resolution branch:

$$\mathcal{P}_{\text{flat}} = \text{Flatten}(\mathcal{P}) \in \mathbb{R}^{B\times CT} \tag{30}$$

$$\mathcal{H}_1 = \text{ReLU}(\text{LN}(\mathcal{W}_r^1 \mathcal{P}_{\text{flat}} + \mathbf{b}_r^1)) \tag{31}$$

Algorithm 1 Motion Guided Feature Modulation (MGFM)

Require: Input tensor $\mathcal{X} \in \mathbb{R}^{B\times C\times T\times H\times W}$
Ensure: Output tensor $\mathcal{Y} \in \mathbb{R}^{B\times C\times T\times H\times W}$
Motion Estimation:
$\mathcal{M}_1 \leftarrow \text{ReLU}(\text{GroupNorm}(\text{Conv3D}(\mathcal{X})))$ $\{C \rightarrow C/2$ channels$\}$
$\mathcal{M}_2 \leftarrow \text{Conv3D}(\mathcal{M}_1)$ $\{C/2 \rightarrow 2$ channels for flow$\}$
Flow Encoding:
$\mathcal{F}_1 \leftarrow \text{ReLU}(\text{GroupNorm}(\text{Conv3D}(\mathcal{M}_2)))$ $\{2 \rightarrow C/4$ channels$\}$
$\mathcal{F}_2 \leftarrow \text{Conv3D}(\mathcal{F}_1)$ $\{C/4 \rightarrow C$ channels$\}$
Feature Modulation:
$\mathcal{X}_{\text{mod}} \leftarrow \mathcal{X} \odot \sigma(\mathcal{F}_2)$ {Element-wise multiplication}
Feature Enhancement:
$\mathcal{X}_{\text{cat}} \leftarrow [\mathcal{X}, \mathcal{X}_{\text{mod}}]$ {Channel-wise concatenation}
$\mathcal{E}_1 \leftarrow \text{ReLU}(\text{GroupNorm}(\text{Conv3D}(\mathcal{X}_{\text{cat}})))$ $\{2C \rightarrow C$ channels$\}$
$\mathcal{E}_2 \leftarrow \text{Conv3D}(\mathcal{E}_1)$ $\{C \rightarrow C$ channels$\}$
Residual Connection:
$\mathcal{Y} \leftarrow \mathcal{E}_2 + \mathcal{X}$

return $\mathcal{Y}$

$$\mathcal{H}_2 = \text{ReLU}(\text{LN}(\mathcal{W}_r^2\mathcal{H}_1 + \mathbf{b}_r^2)) \tag{32}$$

$$\boldsymbol{\alpha} = \text{Softmax}(\mathcal{W}_r^3\mathcal{H}_2 + \mathbf{b}_r^3) \tag{33}$$

where $\mathcal{W}_r^1 \in \mathbb{R}^{256\times CT}$, $\mathbf{b}_r^1 \in \mathbb{R}^{256}$ are the weights and biases of the first fully connected layer, $\mathcal{W}_r^2 \in \mathbb{R}^{128\times 256}$, $\mathbf{b}_r^2 \in \mathbb{R}^{128}$ are the weights and biases of the second fully connected layer, $\mathcal{W}_r^3 \in \mathbb{R}^{R\times 128}$, $\mathbf{b}_r^3 \in \mathbb{R}^R$ are the weights and biases of the third fully connected layer, $\text{LN}(\cdot)$ denotes Layer Normalization, R is the number of temporal resolution branches (default is 3), and $\boldsymbol{\alpha} \in \mathbb{R}^{B\times R}$ represents the predicted weights for each temporal resolution branch

4.3.3. Multi-Resolution Temporal Processing

The input is processed through multiple temporal convolution branches with different kernel sizes and dilation rates:

$$\mathcal{T}_r = \text{ReLU}(\text{GN}(\mathcal{W}_t^r *_{K_r,D_r} \mathcal{X} + \mathbf{b}_t^r)) \tag{34}$$

$$\mathcal{T}_r' = \mathcal{W}_p^r * \mathcal{T}_r + \mathbf{b}_p^r \tag{35}$$

For each resolution branch $r \in \{1, 2, ..., R\}$, where $\mathcal{W}_t^r \in \mathbb{R}^{C\times C\times K_r\times 1\times 1}$ and $\mathbf{b}_t^r \in \mathbb{R}^C$ are the weights and biases of the depthwise temporal convolution, $\mathcal{W}_p^r \in \mathbb{R}^{C\times C\times 1\times 1\times 1}$ and $\mathbf{b}_p^r \in \mathbb{R}^C$ are the weights and biases of the pointwise convolution, $*_{K_r,D_r}$ denotes the convolution operation with kernel size $K_r = 3 + 2r$ and dilation rate $D_r = 2^r$ along the temporal dimension, and $\text{GN}(\cdot)$ denotes Group Normalization with number of groups $\min(32, C)$

4.3.4. Adaptive Fusion

The outputs from each resolution branch are weighted and combined:

$$\mathcal{T}' = \sum_{r=1}^{R} \alpha_r \cdot \mathcal{T}_r' \tag{36}$$

where α_r is the predicted weight for the r-th branch from $\boldsymbol{\alpha}$, reshaped to $\mathbb{R}^{B\times 1\times 1\times 1\times 1}$ for broadcasting, and $\mathcal{T}' \in \mathbb{R}^{B\times C\times T\times H\times W}$ is the weighted sum of the branch outputs

4.3.5. Enhancement with Residual Connection

The combined features are enhanced and a residual connection is added:

$$\mathcal{E}_1 = \text{ReLU}(\text{GN}(\mathcal{W}_e^1 * \mathcal{T}' + \mathbf{b}_e^1)) \tag{37}$$

$$\mathcal{E}_2 = \mathcal{W}_e^2 * \mathcal{E}_1 + \mathbf{b}_e^2 \tag{38}$$

$$\mathcal{Y} = \mathcal{E}_2 + \mathcal{T}' \tag{39}$$

where $\mathcal{W}_e^1 \in \mathbb{R}^{C\times C\times 3\times 3\times 3}$ and $\mathbf{b}_e^1 \in \mathbb{R}^C$ are the weights and biases of the first enhancement convolution, $\mathcal{W}_e^2 \in \mathbb{R}^{C\times C\times 1\times 1\times 1}$ and $\mathbf{b}_e^2 \in \mathbb{R}^C$ are the weights and biases of the second enhancement convolution, $\text{GN}(\cdot)$ denotes Group Normalization with number of groups $\min(32, C)$, and $\mathcal{Y} \in \mathbb{R}^{B\times C\times T\times H\times W}$ is the output of the ATR module

4.4. Transfer Learning Strategies

In this work, 4 different transfer learning strategies are used as shown in Fig. 4. These strategies are elaborated below in detail:

1. **Strategy 1: Early Feature Transfer** This strategy transfers only the Early Feature Extraction components (Conu3D blocks and Motion Guided Feature modules) from the source model trained on simple settings. The rationale is that basic spatio-temporal features extracted from event streams—such as edge patterns, motion onsets, and fundamental temporal dynamics—remain consistent across different complexity levels. These low-level features capture the fundamental nature of event data representation regardless of environmental challenges. The mid-level and deep processing layers are retrained to adapt to the specific complexities of challenging settings.
2. **Strategy 2: Mid-Level Feature Transfer** This approach transfers the Mid-Level Processing components, including the Adaptive Temporal Resolution Module and associated processing blocks. Mid-level features often represent more abstract motion patterns, temporal sequences, and intermediate action representations that may generalize well between simple and challenging settings. The temporal attention and adaptive resolution mechanisms learned from simple scenarios can provide valuable priors for handling varying temporal complexities in challenging environments, while allowing early and deep layers to adapt to the specific characteristics of the target domain.
3. **Strategy 3: Deep Feature Transfer**
Strategy 3 transfers the Deep Feature Processing layers, which encode high-level semantic representations and complex action patterns. These deep features capture sophisticated motion relationships and action semantics that transcend environmental complexity. By transferring these high-level representations, the

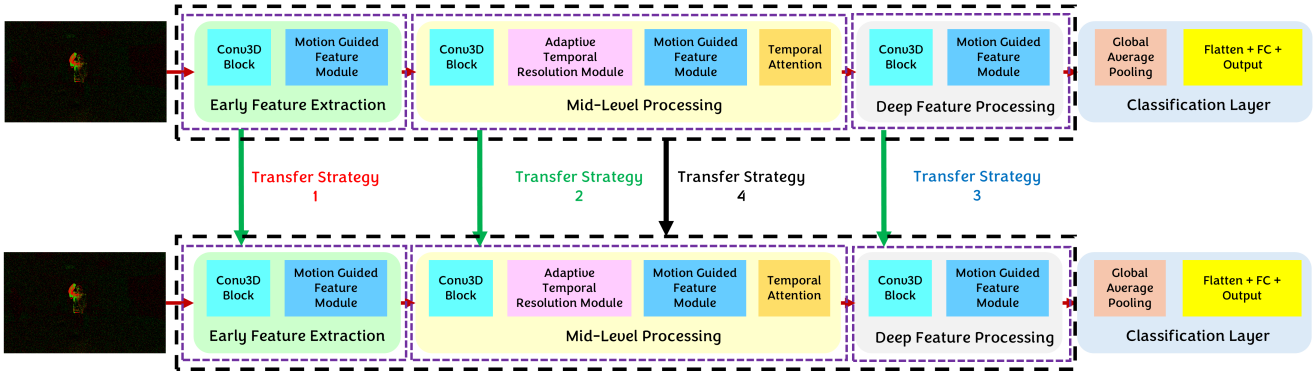

Figure 4: Different Transfer Learning Strategies Applied on the Proposed Model

Algorithm 2 Adaptive Temporal Resolution (ATR)

Require: Input tensor $\mathcal{X} \in \mathbb{R}^{B\times C\times T\times H\times W}$, Number of resolution branches R
Ensure: Output tensor $\mathcal{Y} \in \mathbb{R}^{B\times C\times T\times H\times W}$

Temporal Pooling:
$\mathcal{P} \leftarrow \text{AdaptiveAvgPool3D}(\mathcal{X}, (T, 1, 1))$
$\mathcal{P}_{\text{flat}} \leftarrow \text{Flatten}(\mathcal{P})$
Resolution Prediction:
$\mathcal{H}_1 \leftarrow \text{ReLU}(\text{LayerNorm}(\text{Linear}(\mathcal{P}_{\text{flat}}))) \{CT \rightarrow 256\}$
$\mathcal{H}_2 \leftarrow \text{ReLU}(\text{LayerNorm}(\text{Linear}(\mathcal{H}_1))) \{256 \rightarrow 128\}$
$\boldsymbol{\alpha} \leftarrow \text{Softmax}(\text{Linear}(\mathcal{H}_2)) \{128 \rightarrow R\}$
Multi-Resolution Processing:
for $r = 1$ to R **do**
  $K_r \leftarrow 3 + 2r$ {Increasing kernel sizes: 3, 5, 7, ...}
  $D_r \leftarrow 2^r$ {Increasing dilation rates: 1, 2, 4, ...}
  $P_r \leftarrow (K_r//2) \times D_r$ {Appropriate padding}
  Depthwise Temporal Convolution:
  $\mathcal{T}_r \leftarrow \text{DepthwiseConv3D}(\mathcal{X}, \text{kernel} = (K_r, 1, 1), \text{dilation} = (D_r, 1, 1), \text{padding} = (P_r, 0, 0))$
  $\mathcal{T}_r \leftarrow \text{GroupNorm}(\mathcal{T}_r)$
  $\mathcal{T}_r \leftarrow \text{ReLU}(\mathcal{T}_r)$
  Pointwise Convolution:
  $\mathcal{T}'_r \leftarrow \text{Conv3D}(\mathcal{T}_r, \text{kernel} = (1, 1, 1))$
  $\mathcal{T}'_r \leftarrow \text{GroupNorm}(\mathcal{T}'_r)$
  Apply Branch Weight:
  $\alpha_r \leftarrow \boldsymbol{\alpha}[:, r].\text{reshape}(B, 1, 1, 1, 1)$ {Reshape for broadcasting}
  $\mathcal{T}'_r \leftarrow \mathcal{T}'_r \times \alpha_r$
end for
Branch Fusion:
$\mathcal{T}' \leftarrow \sum_{r=1}^{R} \mathcal{T}'_r$
Enhancement:
$\mathcal{E}_1 \leftarrow \text{ReLU}(\text{GroupNorm}(\text{Conv3D}(\mathcal{T}', \text{kernel} = (3, 3, 3), \text{padding} = (1, 1, 1))))$
$\mathcal{E}_2 \leftarrow \text{Conv3D}(\mathcal{E}_1, \text{kernel} = (1, 1, 1))$
Residual Connection:
$\mathcal{Y} \leftarrow \mathcal{E}_2 + \mathcal{T}'$

return $\mathcal{Y}$

model leverages learned action understanding from simple settings while allowing earlier layers to adapt to the challenging visual conditions (lighting variations, occlusions, noise) present in the target domain.

4. **Strategy 4: Comprehensive Feature Transfer** This strategy performs the most extensive transfer by transferring all three processing stages—Early Feature Extraction, Mid-Level Processing, and Deep Feature Processing. Only the final Classification Layer is retrained for the target domain. This approach assumes that the entire feature hierarchy learned from simple settings provides a robust foundation for challenging scenarios. The comprehensive transfer leverages all levels of learned representations while requiring only task-specific adaptation in the classification head to accommodate potential differences in action categories or decision boundaries between simple and challenging settings.

4.5. Quantization

Following the transfer learning phase, the trained model undergoes quantization to reduce computational complexity and memory requirements for deployment on resource-constrained edge devices. The quantization process transforms the full-precision weights $W \in \mathbb{R}$ and activations $A \in \mathbb{R}$ into their quantized counterparts using uniform quantization. For 8-bit quantization, the quantized weights are computed as:

$$W_q = \text{round}\left(\frac{W - W_{\min}}{S_w}\right) \tag{40}$$

where $S_w = \frac{W_{\max} - W_{\min}}{2^8 - 1}$ represents the weight scaling factor. Similarly, activations are quantized as:

$$A_q = \text{round}\left(\frac{A - A_{\min}}{S_a}\right) \tag{41}$$

with scaling factor $S_a = \frac{A_{\max} - A_{\min}}{2^8 - 1}$. The quantized inference is performed as:

$$Y_q = \text{ReLU}\left(\frac{S_w \cdot S_a}{S_y} \cdot (W_q \cdot A_q) + b_q\right) \tag{42}$$

Table 1
Hyperparameters of the MATRIX-HAR Model used for both Dataset 1 and Dataset 2

Parameter Category	Parameter Name	Value
Training Parameters	Batch Size	16
	Number of Epochs	100
	Initial Learning Rate	0.001
	Weight Decay	1×10^{-6}
	Gradient Clipping Norm	1.0
	Early Stopping Patience	50 epochs
Network Backbone	Main Convolution Filters	128
	Conv Kernel Size	$3 \times 3 \times 3$
	Pooling Method	Max pooling
	Pooling Kernel	$2 \times 2 \times 2$
Classification Head	FC Layer Hidden Dimension	512
	Dropout Rate	0.5
	Output Dimension	Number of classes (50)
Motion Guided Feature Modulation (MGFM)	Number of MGFM Modules	3
	Module Positions	Early, Mid, Late stages
	Motion Conv Filters	[filters/2, 2]
	Motion Conv Kernel Size	$3 \times 3 \times 3$
	Flow Encoder Filters	[filters/4, filters]
	Enhancement Conv Filters	[filters, filters]
	Group Normalization	Min(32, filters) groups
Adaptive Temporal Resolution (ATR)	Number of Resolution Branches	3
	Temporal Kernel Sizes	[3, 5, 7]
	Dilation Rates	[1, 2, 4]
	Resolution Predictor Hidden	[256, 128]
	Predictor Activation	ReLU
	Predictor Normalization	LayerNorm
	Branch Convolution Type	Depthwise + Pointwise
	Branch Normalization	GroupNorm (min(32, filters))
	Dropout Rate	0.3
Temporal Attention	Attention Type	Channel-wise
	Attention Conv Filters	1
	Kernel Size	$1 \times 1 \times 1$
	Activation Function	Sigmoid
Optimization Strategy	Optimizer	Adam
	LR Scheduler	ReduceLROnPlateau
	LR Reduction Factor	0.1
	LR Scheduler Patience	5 epochs
	LR Scheduler Mode	min (validation loss)
Number of Parameters		1.2 M

where S_y is the output scaling factor and b_q is the quantized bias. This post-training quantization approach maintains the learned representations from transfer learning while achieving significant compression ratios, typically reducing model size by 4× and enabling efficient deployment for real-time event-based action recognition.

5. Results and Discussion

This section presents a comprehensive evaluation of the proposed MATRIX-HAR framework across four benchmark datasets, analyzing the effectiveness of individual modules, transfer learning strategies, and quantization impacts. We systematically examine the model's performance through ablation studies, state-of-the-art comparisons, and detailed analysis of learning dynamics to validate the proposed architectural innovations and transfer learning methodologies.

5.1. Ablation Study

The ablation study reveals the distinct contributions of each proposed module across four benchmark datasets, demonstrating the complementary nature of Temporal Attention (TA), Motion Guided Feature Module (MGFM), and

Table 2
Ablation Study of Proposed Model on Benchmark Datasets without Transfer Learning

Modules			HARDVS			PAF			THU-EACT-50			THU-EACT-50 CHL		
TA	MGFM	ATR	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5
✓	×	×	93.9%	94.5%	97.2%	95.1%	96.9%	97.7%	90.1%	93.9%	96.7%	53.1%	71.0%	81.3%
✓	✓	×	94.8%	95.7%	98.8%	96.2%	97.6%	98.8%	91.3%	95.1%	97.4%	54.1%	73.1%	83.2%
✓	✓	✓	97.8%	98.6%	99.7%	97.2%	98.4%	100%	95.7%	99.1%	99.5%	58.4%	78.1%	86.3%

Adaptive Temporal Resolution (ATR) in enhancing event-based action recognition performance. The baseline configuration with only TA achieves strong performance on simpler datasets like HARDVS (93.9% Top-1) and PAF (95.1% Top-1), but shows notable degradation on the challenging THU-EACT-50 CHL dataset (53.1% Top-1), highlighting the limitations of temporal attention alone in complex scenarios. The addition of MGFM provides consistent improvements across all datasets, with gains ranging from 0.9% to 1.2% in Top-1 accuracy for the simpler datasets and 1.0% for the challenging dataset. This demonstrates MGFM's effectiveness in capturing motion-specific features that enhance the discriminative power of learned representations, particularly beneficial for event cameras where motion information is intrinsically encoded in the data stream.

The most significant performance gains are observed when all three modules are integrated, with the complete architecture achieving substantial improvements of 3.9% (HARDVS), 2.1% (PAF), 5.6% (THU-EACT-50), and 5.3% (THU-EACT-50 CHL) in Top-1 accuracy compared to the baseline. The ATR module emerges as the most impactful component, contributing approximately 3.0% improvement on simpler datasets and 4.3% on challenging ones when combined with TA and MGFM. This suggests that adaptive temporal resolution is crucial for handling varying action durations and temporal complexities inherent in event-based data. Notably, the performance gap between simple and challenging datasets (39.4% difference in Top-1 accuracy for the complete model) underscores the difficulty posed by complex environmental conditions, yet the proposed modules collectively provide robustness across varying complexity levels. The consistent improvements across Top-1, Top-2, and Top-5 metrics validate the architectural design's effectiveness in producing well-calibrated confidence scores and meaningful feature hierarchies for event-based action recognition.

5.2. Evaluation of Transfer Strategy

The transfer learning results without quantization demonstrate the varying effectiveness of different knowledge transfer strategies when adapting from simple (THU-EACT-50) to challenging (THU-EACT-50 CHL) event-based action recognition scenarios. All transfer strategies maintain identical performance on the source dataset (95.7% Top-1 accuracy), indicating successful preservation of learned capabilities from the simple settings. However, their performance on the challenging target dataset reveals significant differences in transferability across architectural levels. Strategy 2 (mid-level transfer) shows modest improvements over Strategy 1 (early-level transfer), achieving 61.4% versus 58.9% Top-1 accuracy, suggesting that mid-level temporal processing and adaptive mechanisms provide some robustness to environmental complexities. Strategy 3 (deep-level transfer) performs comparably to Strategy 1 with 59.4% Top-1 accuracy, indicating that high-level semantic features alone may not effectively bridge the domain gap between simple and challenging conditions. Most notably, Strategy 4 (comprehensive transfer) achieves the best target performance with 65.8% Top-1 accuracy, representing a substantial 6.9% improvement over the baseline Strategy 1, demonstrating that transferring the entire feature hierarchy provides the most effective adaptation to challenging environments.

The impact of quantization on transfer learning performance reveals the robustness of the proposed approach under computational constraints. Quantization introduces minimal degradation on the source dataset, with Top-1 accuracy dropping slightly from 95.7% to 95.1% across all strategies, representing only a 0.6% performance loss. This demonstrates the model's resilience to 8-bit quantization while maintaining competitive accuracy on the source domain. On the challenging target dataset, quantization causes marginally larger degradation, with Strategy 4 experiencing a 0.6% drop from 65.8% to 65.2% Top-1 accuracy, while other strategies show similar patterns with 0.5-0.6% decreases. Importantly, the relative performance ranking between strategies remains consistent after quantization, with Strategy 4 continuing to outperform others by significant margins (6.9% over Strategy 1 and 4.4% over Strategy 2 in the quantized setting). The consistent Top-2 and Top-5 improvements across strategies further validate the effectiveness of comprehensive feature transfer, with Strategy 4 achieving 85.1% Top-2 and 94.5% Top-5 accuracy even after quantization. These results demonstrate that the proposed quantization approach successfully maintains the benefits of transfer learning while enabling efficient deployment, making it particularly suitable for resource-constrained edge devices performing real-time event-based action recognition in challenging environments.

The comprehensive analysis of learning dynamics and efficiency metrics reveals the superior adaptation characteristics of different transfer strategies, particularly highlighting Strategy 4's dominance across multiple evaluation criteria. The radar chart analysis demonstrates that Strategy 4 achieves the highest scores in learning speed, final accuracy

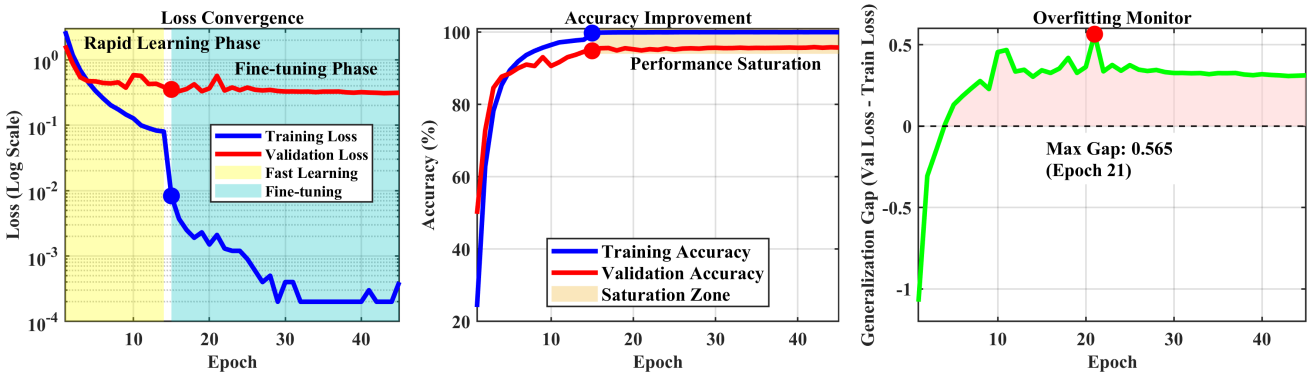

Figure 5: Training and Validation Matrices over the Epochs for the Proposed Model on THU-E-ACT-50 Dataset

Table 3
Comparison of Different Transfer Strategies of the Proposed Model without Quantisation from THU-E-ACT-50 to THU-E-ACT-50 CHL

TS	THU-EACT-50			THU-EACT-50 CHL		
	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5
1	95.7%	99.1%	99.5%	58.9%	78.9%	87.1%
2	95.7%	99.1%	99.5%	61.4%	80.4%	89.2%
3	95.7%	99.1%	99.5%	59.4%	79.2%	88.4%
4	95.7%	99.1%	99.5%	65.8%	85.5%	94.9%

Table 4
Comparison of Different Transfer Strategy's of the Proposed Model with Quantisation from THU-E-ACT-50 to THU-E-ACT-50 CHL

TS	THU-EACT-50			THU-EACT-50 CHL		
	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5
1	95.1%	98.6%	99.3%	58.3%	78.5%	86.8%
2	95.1%	98.6%	99.3%	60.8%	79.9%	88.9%
3	95.1%	98.6%	99.3%	58.8%	78.8%	88.1%
4	95.1%	98.6%	99.3%	65.2%	85.1%	94.5%

(65.8%), and stability, while maintaining competitive efficiency, establishing it as the most balanced and effective approach. The relative performance gains clearly illustrate the progressive improvement from Strategy 1 (9.5%) to Strategy 4 (22.3%), representing a substantial 12.8% advantage of comprehensive transfer over early-level transfer. Convergence efficiency analysis shows that transfer learning strategies require fewer epochs to reach 90% of final accuracy (7-10 epochs) compared to training from scratch (10 epochs), with Strategies 1 and 2 demonstrating the fastest convergence. The learning velocity and acceleration curves provide deeper insights into the optimization dynamics, where the original model exhibits extreme initial learning rates (peak of 32 accuracy/epoch) followed by dramatic deceleration, indicating potential instability and overfitting. In contrast, all transfer strategies demonstrate more controlled and stable learning patterns with peak velocities around 7-9 accuracy/epoch and smoother acceleration profiles, suggesting that pre-trained features provide better optimization landscapes

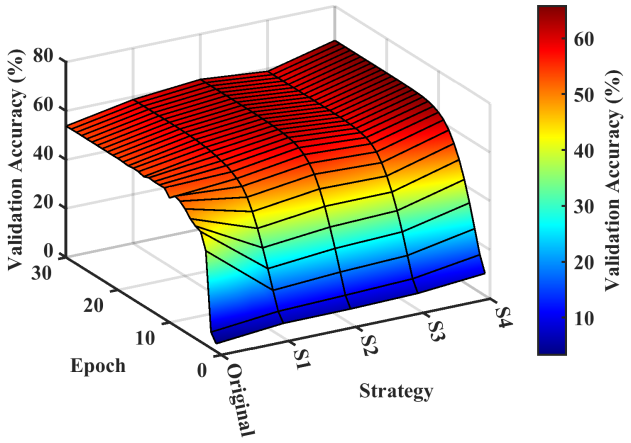

Figure 6: Comparison of Validation Accuracies over the Epochs for Different Transfer Learning Strategies as compared to original model

and improved training stability, ultimately leading to more reliable convergence to superior performance levels.

5.3. Comparison with State of the Art

The comprehensive comparison with state-of-the-art methods across four benchmark datasets demonstrates the superior performance of MATRIX-HAR in event-based human action recognition tasks. On the HARDVS dataset, MATRIX-HAR achieves remarkable Top-1 accuracy of 97.82%, outperforming the previous best method EventCrab by 0.71% and establishing new state-of-the-art results across all evaluation metrics (Top-1, Top-2, and Top-5). The performance gap becomes even more pronounced when compared to conventional video understanding methods, with MATRIX-HAR surpassing traditional approaches like SlowFast, ACTION-Net, and TSCFormer by substantial margins exceeding 44%. This significant improvement highlights the importance of specialized architectures designed for event camera data processing. Similarly, on the PAF dataset, MATRIX-HAR achieves 97.23% Top-1 accuracy, surpassing EventCrab by 0.74% and maintaining perfect Top-5 accuracy of 100%. The consistent superiority over both

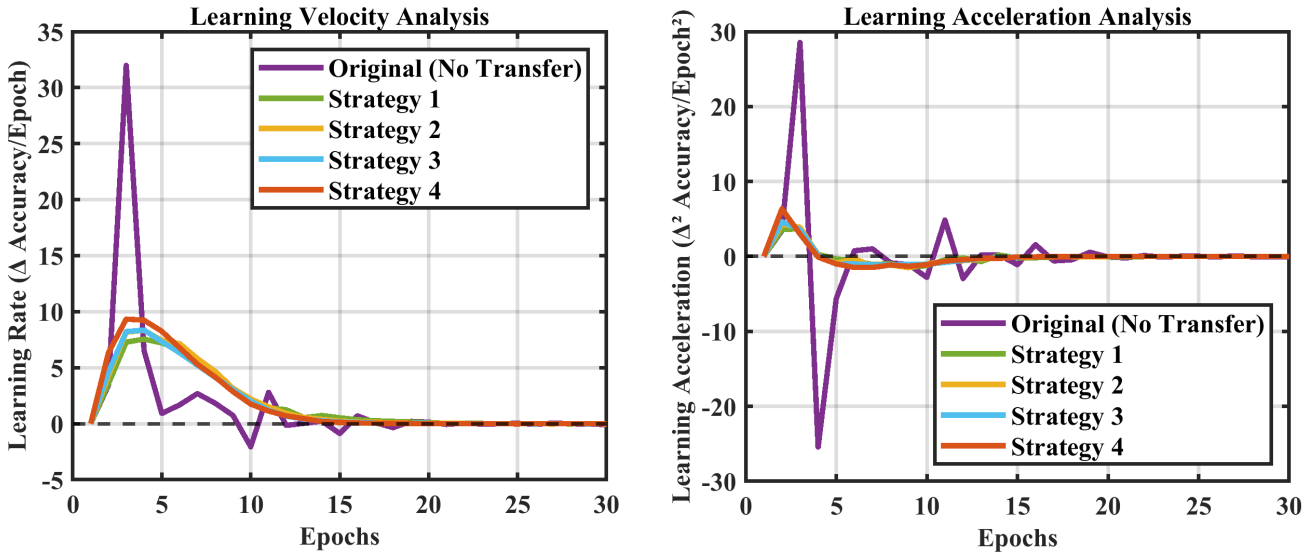

Figure 7: Learning Dynamics Analysis of the Different Transfer Strategies

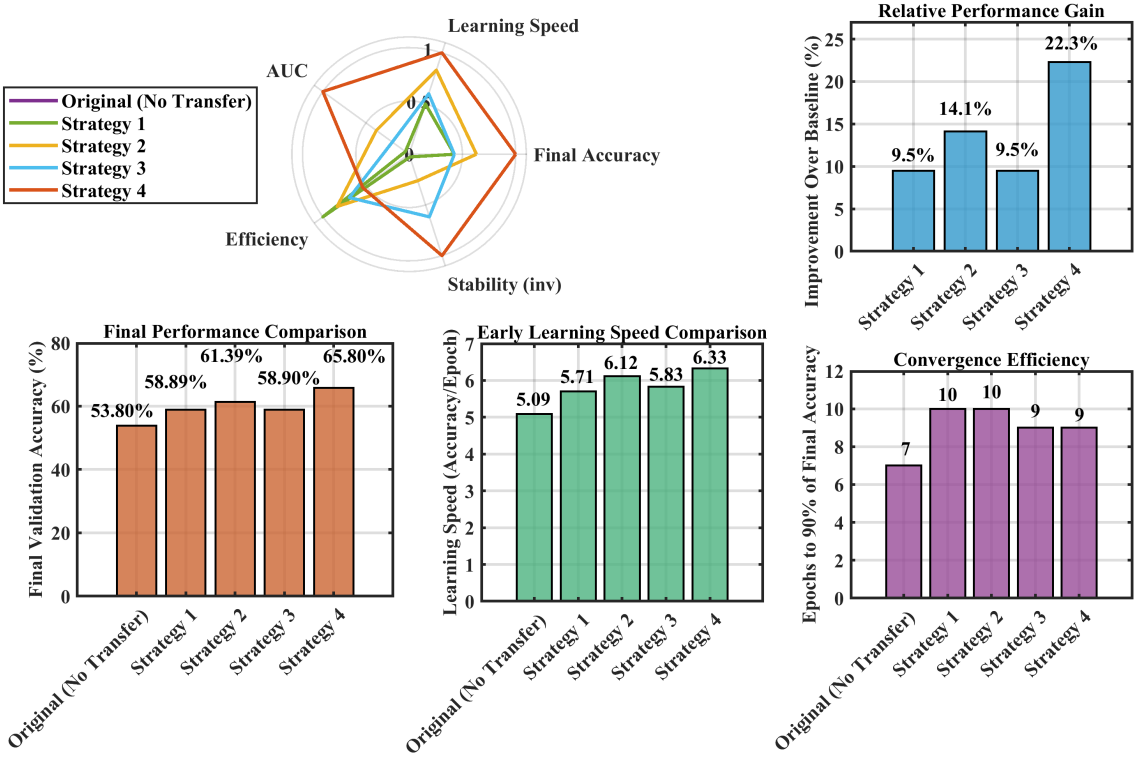

Figure 8: Evaluation Comparison of the Different Transfer Learning Strategies with the Original Model

SNN-based approaches (HMAX SNN, Motion SNN) and transformer-based methods (Swin-T) validates the effectiveness of the proposed motion-guided feature modules and adaptive temporal resolution mechanisms.

The most compelling evidence of MATRIX-HAR's robustness emerges from its performance on the challenging datasets THU-E-ACT-50 and THU-E-ACT-50 CHL, where environmental complexities significantly impact recognition accuracy. On THU-E-ACT-50, MATRIX-HAR achieves 95.73% Top-1 accuracy, representing a substantial 3.03% improvement over the previous best method EV-ACT. More remarkably, on the challenging THU-E-ACT-50 CHL dataset, MATRIX-HAR demonstrates exceptional robustness with 65.8% Top-1 accuracy, representing a significant 7.3% improvement over EV-ACT and nearly doubling the performance of HMAX SNN (32.7%). The dramatic performance degradation experienced by competing methods when transitioning from simple to challenging conditions (EV-ACT

Table 5
Comparison with State of the Art Models

Dataset	Model	Accuracy(%)		
		Top-1	Top-2	Top-5
HARDVS	SlowFast	46.54	\	54.76
	ACTION-Net	46.85	\	56.19
	R2Plus1D	49.06	\	56.43
	ResNet18	49.2	\	56.09
	TAM	50.41	\	57.99
	C3D	50.52	\	56.14
	ESTF	51.22	\	57.53
	Video-SwinTrans	51.91	\	59.11
	TSM	52.63	\	60.56
	TSCFormer	53.04	\	62.67
	ExACT	90.1	\	96.69
	EventCrab	97.11	\	99.48
	MATRIX-HAR (Our)	**97.82(+0.71)**	**98.63**	**99.71(+0.23)**
PAF	HMAX SNN	55	\	\
	STCA	71.2	\	\
	Motion SNN	78.1	\	\
	MST	88.21	\	\
	Swin-T(BN)	90.14	\	\
	EV-ACT	92.6	\	\
	ExACT	94.83	\	98.28
	EventCrab	96.49	\	100
	MATRIX-HAR (Our)	**97.23(+0.74)**	**98.41**	100(+0)
THU-E-ACT-50	HMAX SNN	65.1	\	\
	Motion SNN	85.3	\	\
	EV-ACT	92.7	98.4	99.2
	MATRIX-HAR (Our)	**95.73(+3.03)**	**99.11(+0.71)**	**99.5(+0.3)**
THU-E-ACT-50 CHL	HMAX SNN	32.7	\	\
	Motion SNN	47.3	\	\
	EV-ACT	58.5	78.6	86
	MATRIX-HAR (Our)	**65.8(+7.3)**	**85.5(+6.9)**	**94.9(+8.9)**

drops from 92.7% to 58.5%) contrasts sharply with MATRIX-HAR's more graceful degradation (from 95.73% to 65.8%), indicating superior adaptability to environmental variations. Across all datasets, MATRIX-HAR consistently achieves the highest Top-2 and Top-5 accuracies, demonstrating not only superior classification performance but also better confidence calibration and feature representation quality, making it particularly suitable for real-world deployment scenarios where environmental conditions are unpredictable.

6. Conclusion

In this work, we have presented MATRIX-HAR, a comprehensive framework for event-based human action recognition that effectively addresses the challenges of spatio-temporal feature extraction and domain adaptation in varying environmental conditions. Our proposed architecture integrates three key innovations: the Adaptive Temporal Resolution (ATR) module for dynamic temporal processing, the Motion Guided Feature Module (MGFM) for enhanced motion-aware feature extraction, and a systematic hierarchical transfer learning approach that enables robust adaptation from simple to challenging scenarios. Through extensive experiments on four benchmark datasets, we have demonstrated that MATRIX-HAR achieves state-of-the-art performance, surpassing existing methods by substantial margins, particularly excelling in challenging environmental conditions where traditional approaches face significant degradation. The systematic evaluation of four transfer learning strategies reveals that comprehensive feature transfer (Strategy 4) provides the most effective adaptation mechanism, achieving 22.3% relative performance improvement over baseline methods while maintaining computational efficiency through quantization. Our ablation studies confirm the complementary nature of the proposed modules, with each component contributing to the overall performance enhancement. The learning dynamics analysis further validates the superior convergence properties and training stability of transfer learning approaches compared to training from scratch. Importantly, the quantization-aware deployment demonstrates the practical viability of our framework for resource-constrained edge devices, achieving 4× compression ratios with minimal performance degradation.

Acknowledgement

This research is supported by the Artificial Intelligence, Biomechatronics, and Collaborative Robotics research group at the Top Research Center Mechatronics (TRCM), University of Agder (UiA), Norway.

References

[1] F. Sanfilippo, M. H. Zafar, T. Wiley, and F. Zambetta, "From caged robots to high-fives in robotics: Exploring the paradigm shift from human–robot interaction to human–robot teaming in human–machine interfaces," *Journal of Manufacturing Systems*, vol. 78, pp. 1–25, 2025.

[2] Y. Zhang, K. Ding, J. Hui, S. Liu, W. Guo, and L. Wang, "Skeleton-rgb integrated highly similar human action prediction in human–robot collaborative assembly," *Robotics and Computer-Integrated Manufacturing*, vol. 86, p. 102 659, 2024.

[3] D. Regazzoni, G. De Vecchi, and C. Rizzi, "Rgb cams vs rgb-d sensors: Low cost motion capture technologies performances and limitations," *Journal of Manufacturing Systems*, vol. 33, no. 4, pp. 719–728, 2014.

[4] G. Gallego, T. Delbrück, G. Orchard, *et al.*, "Event-based vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 154–180, 2020.

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

[6] S. U. Innocenti, F. Becattini, F. Pernici, and A. Del Bimbo, "Temporal binary representation for event-based action recognition," in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 10 426–10 432.

[7] S. A. Baby, B. Vinod, C. Chinni, and K. Mitra, "Dynamic vision sensors for human activity recognition," in *2017 4th IAPR Asian conference on pattern recognition (ACPR)*, IEEE, 2017, pp. 316–321.

[8] X. Clady, J.-M. Maro, S. Barré, and R. B. Benosman, "A motion-based feature for event-based pattern recognition," *Frontiers in neuroscience*, vol. 10, p. 594, 2017.

[9] G. Layher, T. Brosch, and H. Neumann, "Real-time biologically inspired action recognition from key poses using a neuromorphic architecture," *Frontiers in neurorobotics*, vol. 11, p. 13, 2017.

[10] 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 *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2019, pp. 1–5.

[11] P. Pansuriya, N. Choksi, D. Patel, and S. Vahora, "Human activity recognition with event-based dynamic vision sensor using deep recurrent neural network," *International Journal of Advanced Science and Technology*, vol. 29, no. 4, pp. 9084–9091, 2020.

[12] C. Huang, "Event-based timestamp image encoding network for human action recognition and anticipation," in *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2021, pp. 1–9.

[13] C. Liu, X. Qi, E. Y. Lam, and N. Wong, "Fast classification and action recognition with event-based imaging," *IEEE access*, vol. 10, pp. 55 638–55 649, 2022.

[14] G. Chen, S. Qu, Z. Li, *et al.*, "Neuromorphic vision-based fall localization in event streams with temporal–spatial attention weighted network," *IEEE transactions on cybernetics*, vol. 52, no. 9, pp. 9251–9262, 2022.

[15] S. S. Prasad, N. K. Mehta, H. Kumar, A. Banerjee, S. Saurav, and S. Singh, "Hybrid snn-based privacy-preserving fall detection using neuromorphic sensors," in *Proceedings of the Fourteenth Indian Conference on Computer Vision, Graphics and Image Processing*, 2023, pp. 1–9.

[16] F. J. Moreno-Rodríguez, V. J. Traver, F. Barranco, M. Dimiccoli, and F. Pla, "Visual event-based egocentric human action recognition," in *Iberian Conference on Pattern Recognition and Image Analysis*, Springer, 2022, pp. 402–414.

[17] C. Plizzari, M. Planamente, G. Goletto, *et al.*, "E2 (go) motion: Motion augmented event stream for egocentric action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 19 935–19 947.

[18] H. Ren, Y. Zhou, H. Fu, Y. Huang, R. Xu, and B. Cheng, "Ttpoint: A tensorized point cloud network for lightweight action recognition with event cameras," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 8026–8034.

[19] M. Cao, X. Shu, J. Zhang, R. Yan, Z. Li, and J. Tang, "Eventcrab: Harnessing frame and point synergy for event-based action recognition and beyond," *arXiv preprint arXiv:2411.18328*, 2024.

[20] X. Wang, Z. Wu, B. Jiang, *et al.*, "Hardvs: Revisiting human activity recognition with dynamic vision sensors," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 5615–5623.

[21] Y. Gao, J. Lu, S. Li, Y. Li, and S. Du, "Hypergraph-based multi-view action recognition using event cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[22] C. Plou, N. Gallego, A. Sabater, *et al.*, "Eventsleep: Sleep activity recognition with event cameras," *arXiv preprint arXiv:2404.01801*, 2024.

[23] L. Cultrera, F. Becattini, L. Berlincioni, C. Ferrari, and A. Del Bimbo, "Spatio-temporal transformers for action unit classification with event cameras," *arXiv preprint arXiv:2410.21958*, 2024.

[24] Q. Liu, D. Xing, H. Tang, D. Ma, and G. Pan, "Event-based action recognition using motion information and spiking neural networks.," in *IJCAI*, 2021, pp. 1743–1749.

[25] S.-C. Yang, M.-L. Wu, and Y.-K. Wang, "Human activity recognition with spiking neural network," in *2022 IEEE 11th Global Conference on Consumer Electronics (GCCE)*, IEEE, 2022, pp. 482–483.

[26] S. Bian and M. Magno, "Evaluating spiking neural network on neuromorphic platform for human activity recognition," in *Proceedings of the 2023 ACM International Symposium on Wearable Computers*, 2023, pp. 82–86.

[27] H. Ren, Y. Zhou, Y. Huang, *et al.*, "Spikepoint: An efficient point-based spiking neural network for event cameras action recognition," *arXiv preprint arXiv:2310.07189*, 2023.

[28] X. Lin, M. Liu, and H. Chen, "Spike-har++: An energy-efficient and lightweight parallel spiking transformer for event-based human action recognition," *Frontiers in Computational Neuroscience*, vol. 18, p. 1 508 297, 2024.

[29] S. Miao, G. Chen, X. Ning, *et al.*, "Neuromorphic vision datasets for pedestrian detection, action recognition, and fall detection," *Frontiers in neurorobotics*, vol. 13, p. 38, 2019.

[30] Y. Gao, J. Lu, S. Li, *et al.*, "Action recognition and benchmark using event cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.