BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 10/13 pp914–932
DOI: 10.26599/BDMA.2025.9020003
Volume 8, Number 4, August 2025

Action Recognition in Real-World Ambient Assisted Living Environment

Vincent Gbouna Zakka*, Zhuangzhuang Dai, and Luis J. Manso

Abstract: The growing ageing population and their preference to maintain independence by living in their own homes require proactive strategies to ensure safety and support. Ambient Assisted Living (AAL) technologies have emerged to facilitate ageing in place by offering continuous monitoring and assistance within the home. Within AAL technologies, action recognition plays a crucial role in interpreting human activities and detecting incidents like falls, mobility decline, or unusual behaviours that may signal worsening health conditions. However, action recognition in practical AAL applications presents challenges, including occlusions, noisy data, and the need for real-time performance. While advancements have been made in accuracy, robustness to noise, and computation efficiency, achieving a balance among them all remains a challenge. To address this challenge, this paper introduces the Robust and Efficient Temporal Convolution network (RE-TCN), which comprises three main elements: Adaptive Temporal Weighting (ATW), Depthwise Separable Convolutions (DSC), and data augmentation techniques. These elements aim to enhance the model's accuracy, robustness against noise and occlusion, and computational efficiency within real-world AAL contexts. RE-TCN outperforms existing models in terms of accuracy, noise and occlusion robustness, and has been validated on four benchmark datasets: NTU RGB+D 60, Northwestern-UCLA, SHREC'17, and DHG-14/28. The code is publicly available at: https://github.com/Gbouna/RE-TCN.

Key words: Ambient Assisted Living (AAL); action recognition; occlusion robust; noise robust; computational efficiency

1 Introduction

According to the United Nations, the number of individuals aged 65 years or older is projected to double by 2050, reaching approximately 1.5 billion worldwide[1]. This demographic shift presents significant challenges for healthcare systems, economics, and society at large[2].

Despite these challenges, most older adults prefer to age in place, desiring to live independently in their own homes rather than relocating to assisted living facilities or nursing homes[3]. However, enabling ageing in place requires proactive measures to ensure safety and support[4], especially as older adults become more susceptible to health risks, such as falls —which are the leading cause of injury-related deaths. In this context, Ambient Assisted Living (AAL) technologies support ageing in place by providing continuous monitoring and assistance within the home environment[5, 6]. Among the various components of AAL systems, action recognition plays a crucial role,

• Vincent Gbouna Zakka and Zhuangzhuang Dai are with School of Computer Science and Digital Technologies, Aston University, Birmingham, B4 7ET, UK. E-mail: vzakk22@aston.ac.uk; dai1@aston.ac.uk.
• Luis J. Manso is with Applied Artificial Intelligence and Robotics Department, Aston University, Birmingham, B4 7ET, UK. E-mail: l.manso@aston.ac.uk.
* To whom correspondence should be addressed.
Manuscript received: 2024-11-16; revised: 2024-12-13; accepted: 2025-01-06

enabling the system to detect and classify human actions, including events, such as falls, mobility decline, or abnormal behaviours, that may indicate health deterioration[7].

Different sensing technologies are common in AAL systems. Wearable devices, such as accelerometers and gyroscopes, are used to monitor movements and detect falls[6, 8, 9]. While wearables provide continuous monitoring, they rely heavily on user compliance causing discomfort, which can be problematic for older individuals, especially those with cognitive impairments[8]. Audio-based systems can monitor sound patterns and movement within the home[10, 11]. However, the accuracy of these systems is degraded in the presence of background noise[11]. Recently, computer vision-based approaches have gained popularity for action recognition in AAL due to their ability to capture rich visual data of human movements and interactions[12].

While many vision-based approaches rely on traditional RGB video data for action recognition, this raises privacy concerns due to the intrusive nature of capturing full-colour video of individuals in their homes[12]. Moreover, RGB-based systems are computationally expensive, requiring significant processing power to handle large volumes of data generated by continuous video streams[13].

A promising alternative is skeleton-based action recognition, which uses skeleton data extracted from images to represent human motion rather than using raw image data[14, 15]. This approach offers several advantages in the context of AAL systems. First, skeleton-based data preserves visual privacy by abstracting the human figure into a set of joints, eliminating the need to capture identifiable features such as facial details or body appearance[16, 17]. Furthermore, skeleton data is compact and lightweight, making it computationally more efficient to process than RGB video. This is an advantage for real-time action recognition tasks in resource-constrained environments[18–21].

However, skeleton-based action recognition faces several challenges in real-world AAL settings as shown in Fig. 1. One of the primary issues is dealing with occlusions, where body parts may be hidden from view due to obstacles or suboptimal camera placement[22]. Another challenge is noisy data resulting from camera inaccuracies or motion artefacts[23]. To address these challenges, various methods have been proposed, focusing on enhancing the robustness[22–24], accuracy[25], and efficiency[26, 27] of skeleton-based systems. While progress has been made in specific areas, such as robustness, accuracy, or real-time performance, a solution that balances all three remains a challenge.

This paper proposes Robust and Efficient Temporal Convolution network (RE-TCN) that builds on a state-of-the-art model Temporal Decoupling Graph Convolution Neural network (TD-GCN)[28] by incorporating three key components: Adaptive Temporal Weighting (ATW), Depthwise Separable Convolutions (DSC) and data augmentation techniques. These components are designed to improve both the accuracy, robustness to noise and occlusion, and computational efficiency of the proposed model in a real-world AAL environment. ATW enhances the model's ability to focus on the most informative frames in an action sequence. It dynamically assigns different levels of importance to each frame, allowing the model to prioritise key moments in the sequence. DSC decomposes the convolution into depthwise and pointwise convolutions, significantly reducing the number of parameters and operations required to process the input skeleton data. The data augmentation techniques were designed to enhance the model's robustness in real-world environments.

We conduct extensive experiments on four benchmark datasets: NTU, RGB+D 60[29],

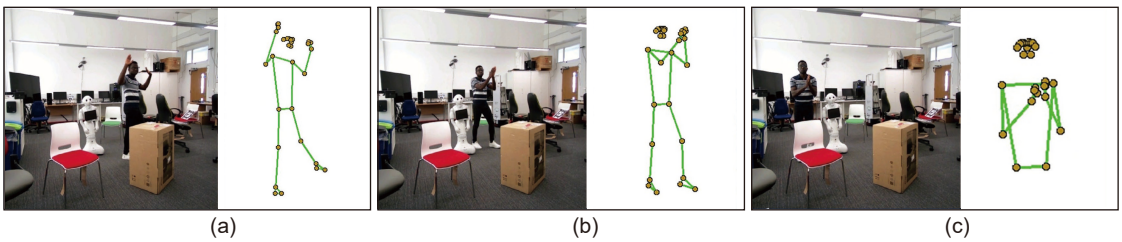


Fig. 1 Challenges in a real-world environment, (a) represents data with noise and occlusion, (b) represents relatively clean data, and (c) represents data with occlusion.

Northwestern-UCLA[30], SHREC17[31], and DHG-14/28[32] to evaluate the effectiveness of the proposal. RE-TCN outperforms existing methods in terms of computational efficiency, accuracy, and robustness to noise and occlusion.

2 Related Work

2.1 Action recognition in AAL

There are two common approaches to Human Activity Recognition (HAR), rule-based and data-driven. Rule-based approaches employ thresholds for triggering alerts of potentially harmful or dangerous events[33]. The second approach leverages machine learning algorithms for HAR[34]. These methods can learn complex patterns from data, enabling more accurate recognition of activities. These methods are implemented using various sensors, generally categorised into wearable and non-wearable sensors.

Inertial sensors are the most common wearable sensors for HAR in ALL applications. These include accelerometers, gyroscopes, and magnetometers[6, 8, 9]. Wearable devices are favoured for their mobility, portability, and accessibility. Numerous studies have utilised wearable devices to detect falls and alert caregivers[35, 36]. However, this approach requires users to constantly wear the devices, which may cause discomfort. Forgetting to wear it can negate the purpose of monitoring, compromising the system's effectiveness.

Non-wearable sensor solutions for HAR involve devices or systems capable of detecting and analysing human activities without direct attachment to the body. Examples include radio-frequency-based systems[37, 38] and, increasingly, vision-based methods[12].

However, vision-based solutions have drawbacks, such as limited field of view, sensitivity to environmental factors like lightning conditions and cluttered backgrounds, and privacy concerns[12, 39]. They also suffer from subject occlusion, which occurs when parts of the subject's body are hidden or obscured by other objects or body parts within the room, leading to incomplete or inaccurate tracking of movements[39].

2.2 Skeleton-based action recognition

Skeleton-based action recognition approaches have gained attention owing to their lower computational complexity compared to processing RGB data. This has led to the development of methods aimed at enhancing action recognition performance.

Graph Convolutional Neural Networks (GCN) have become a fundamental framework in skeleton-based action recognition thanks to their ability to model the spatial and temporal dynamics of human joints effectively. One of the pioneering works is the Spatial-Temporal Graph Convolutional Network (ST-GCN) introduced in Ref. [40], which captures spatial and temporal features by leveraging the graph structure of the skeleton data.

To enhance GCNs' representational ability, several methods have focused on adaptively learning the graph structure[41–50]. These methods employ adaptive GCNs that dynamically learn the graph's topology.

Other approaches leverage attention mechanisms and transformer, and have integrated them into GCN frameworks to improve their ability to capture significant relationships[51–53]. These methods utilise attention mechanisms within graph convolutions to identify key relationships in the data.

Alternative approaches aim to enhance GCN performance by refining the input representation to make it more informative[41, 44, 49]. These works define inputs in terms of bones and motion by preprocessing existing joint data. These methods aim to deepen the network's understanding of underlying action dynamics by providing richer features.

Despite these advancements, few methods address the practical constraints of deploying these systems in real-world environments. Many existing works are unsuitable for real-world deployment, particularly given the computational limitations of edge devices and the cluttered environments in real-world settings that introduce noise and occlusions into the data.

To address some of the real-world challenges, some approaches have concentrated on designing efficient architectures to improve real-time performance[25–27, 54, 55]. Others have focused on mitigating the effects of noise and occlusion in real-world environments by proposing methods to improve performance under these challenging conditions[22–24, 56, 57].

This paper aligns with these efforts by developing an efficient architecture that satisfies the real-time requirements of devices with limited computational resources. Additionally, we address the challenges posed by noise and occlusion in real-world environments to achieve robust performance in

practical applications. Notably, our approach aims to achieve these objectives without sacrificing accuracy.

3 Method

The architecture of RE-TCN is depicted in Fig. 2. The model aims to enhance accuracy, robustness to noise and occlusion, and computational efficiency. The contributions of the method are threefold: an adaptive temporal weighting mechanism, depthwise separable convolution and data augmentation. Details of the implementation are explained below. The objective of human action recognition is to learn a function $f : X \rightarrow y$ that maps the spatio-temporal skeletal data X to an action label y by capturing spatial dependencies between joints and temporal progression across frames. Input data are denoted by X, with dimensions N, C, T, and V, where N is the batch size, C is the number of channels (features), T is the number of frames, and V is the number of joints.

3.1 ATW

The ATW mechanism is developed to dynamically assign different levels of importance to frames within an action sequence. This section outlines the implementation and the design decisions made during its development to optimise computational efficiency.

3.1.1 Implementation of ATW

ATW computes an attention weight, highlighting the importance of different frames for each input sequence. To compute these weights, ATW first collapses the joint dimension by computing the average over joints,

$$X_{\text{proj}} = \frac{1}{V} \sum_{v=1}^{V} X\,[:, :, :, :v] \tag{1}$$

where $X_{\text{proj}} \in \mathbf{R}^{N \times C \times T}$. This step reduces the spatial complexity and focuses on the temporal aspect of the features.

To efficiently compute weight, the temporal feature map is projected into lower-dimensional space using a 1×1 convolution,

$$X_{\text{red}} = \text{Conv1}(X_{\text{proj}}) \tag{2}$$

where $X_{\text{red}} \in \mathbf{R}^{N \times \frac{C}{n} \times T}$. The convolution operation in this step is designed to serve two purposes. First, it reduces the number of channels by a factor of n, enabling more efficient computation while preserving temporal information. Secondly, the convolution operation acts as a feature transformation by combining the information from the temporal features within each channel.

The reduced temporal feature map is restored to its original number of channels using another 1×1 convolution,

$$X_{\text{restored}} = \text{Conv2}\,(X_{\text{red}}) \tag{3}$$

where $X_{\text{restored}} \in \mathbf{R}^{N \times C \times T}$. This step further transforms the intermediate features and restores the dimensionality of the temporal feature map from C/n back to C, ensuring that the feature representation aligns with the original number of channels before weight is applied.

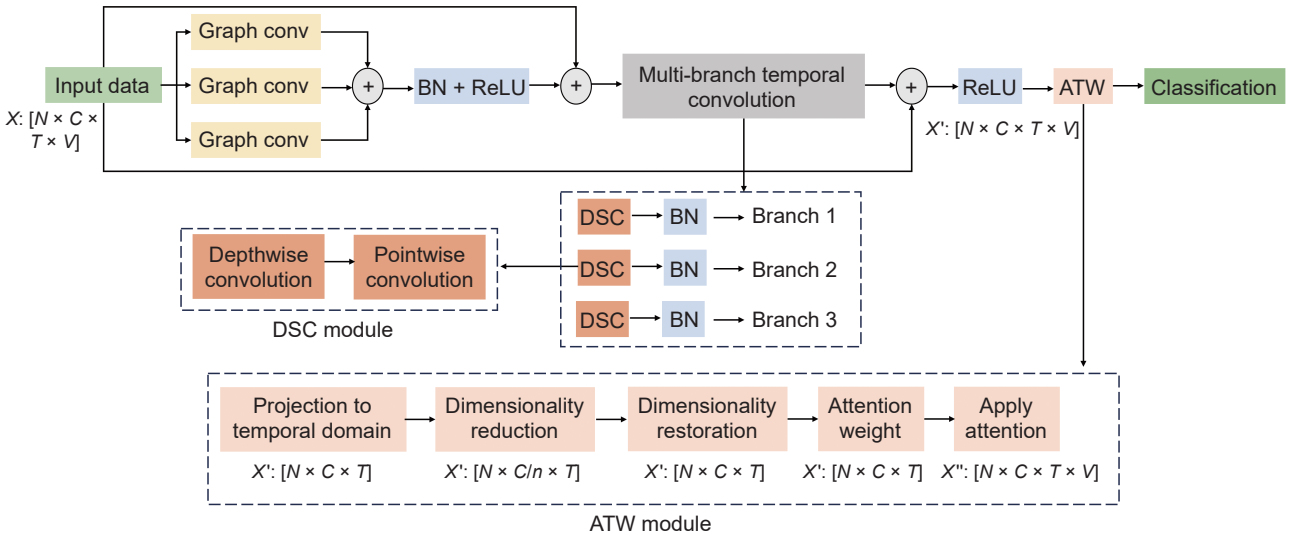


Fig. 2 Architecture of the proposed RE-TCN: Graph convolution is first applied to the skeleton sequences. The output is then passed to the multi-branch temporal convolution, followed by the ATW mechanism, and finally to the classification module for action recognition.

Next, a softmax function is applied across the temporal dimension to compute the attention weights. The softmax operation ensures that the assigned weights sum up to 1,

$$\alpha_t = \frac{\exp\left(X_{\text{restored},\,t}\right)}{\sum_{t=1}^{T} \exp\left(X_{\text{restored},\,t}\right)} \tag{4}$$

where $\alpha_t \in \mathbf{R}^{N\times C\times T}$ are the learned attention weights.

Finally, the original input tensor is weighted by the attention score α_t along the temporal dimension,

$$X' = X \cdot \alpha_t \tag{5}$$

where $X' \in \mathbf{R}^{N\times C\times T\times V}$ is the output of the ATW. Here, the learned attention weights are applied across the temporal dimension, scaling each frame according to its relative importance.

In comparison to the TD-GCN[28], the convolution operation applies convolutions across the temporal dimension as follows:

$$X_{\text{conv}} = \text{Conv}\left(X\right) \tag{6}$$

where each frame is processed uniformly, leading to potential information loss in important frames. In the proposed approach, the temporal convolutions are complemented by ATW which assigns higher importance to informative frames. The difference lies in the introduction of the ATW mechanism that dynamically scales each frame.

$$X_{\text{att}} = X \cdot \alpha_t \tag{7}$$

This ensures that the model learns to focus on the most relevant temporal features, enhancing its ability to capture crucial moments in the action sequence.

3.1.2 Design choices in ATW

The ATW mechanism is designed to achieve computational efficiency while maintaining modelling capacity. Inspired by the depthwise separable convolution process, which breaks down convolution into a two-step operation, ATW's core design replaces a single convolution with $(C_{\text{in}} \to C_{\text{out}})$ channels with two-step convolutions featuring an intermediate dimensionality reduction: $(C_{\text{in}} \to C_{\text{mid}} \to C_{\text{out}})$. This approach divides the convolution operation into two steps. The first convolution operation $C_{\text{in}} \to C_{\text{mid}}$ reduces the channels from C_{in} to an intermediate C_{mid} where $C_{\text{mid}} < C_{\text{in}}$, while the second convolution operation restores the channels back from C_{mid} to C_{out}, where $C_{\text{out}} = C_{\text{in}}$. Here, C_{in}, C_{out}, and C_{mid} are the numbers of input, intermediate, and output channels.

The reduction in computational cost is significant. For instance, the computational cost for a 1×1 convolution operation can be expressed as

$$X_{\text{Cost}} = N \times C_{\text{out}} \times C_{\text{in}} \times T \times V \tag{8}$$

The cost for the first and second convolution operations can therefore be expressed as

$$\text{Cost}_{\text{first}} = N \times C_{\text{mid}} \times C_{\text{in}} \times T \times V \tag{9}$$

$$\text{Cost}_{\text{second}} = N \times C_{\text{out}} \times C_{\text{mid}} \times T \times V \tag{10}$$

The total computational cost for this two-step process is

$$\begin{aligned}\text{Cost}_{\text{total}} =& N \times C_{\text{mid}} \times C_{\text{in}} \times T \times V + \\ & N \times C_{\text{out}} \times C_{\text{mid}} \times T \times V\end{aligned} \tag{11}$$

For comparison, a single convolution operation from $C_{\text{in}} \to C_{\text{out}}$ requires

$$\text{Cost}_{\text{single}} = N \times C_{\text{out}} \times C_{\text{in}} \times T \times V \tag{12}$$

Using two convolutions with an intermediate reduction results in evident computational savings. The ratio of computational cost between the two-step process and the single convolution is

$$\frac{\text{Cost}_{\text{total}}}{\text{Cost}_{\text{single}}} = \frac{C_{\text{mid}} \times C_{\text{in}} + C_{\text{out}} \times C_{\text{mid}}}{C_{\text{out}} \times C_{\text{in}}} \tag{13}$$

For $C_{\text{mid}} << C_{\text{in}}$, this ratio becomes much smaller, demonstrating that the two-step design is significantly more efficient.

3.2 Depthwise separable convolutions

TD-GCN[28] employs convolutions across both temporal and spatial dimensions, enabling it to capture spatio-temporal features effectively and thereby improving the understanding of temporal dynamics and spatial relationships between joints. Despite its strengths, the convolution operation in TD-GCN is computationally intensive and demands a relatively high number of parameters. To reduce the computational cost and improve the model efficiency without sacrificing accuracy, the proposed approach decomposes the convolution operation of TD-GCN into two stages; depthwise and pointwise convolution. The depthwise convolution performs a convolution operation for each input channel (feature), allowing for spatial feature extraction independently on each channel. In contrast, the pointwise convolution uses a 1×1 kernel to combine the outputs of the depthwise convolution across channels.

3.2.1 Convolution operation process

Given an input data $X \in \mathbf{R}^{N \times C_{\text{in}} \times T \times V}$, TD-GCN[28] applies a set of filters $W \in \mathbf{R}^{C_{\text{out}} \times C_{\text{in}} \times K_h \times K_w}$, where $K_h \times K_w$ is the filter size. The output of the convolution operation is computed as

$$Y_{\text{TD-GCN}} = W \times X \tag{14}$$

where $Y_{\text{TD-GCN}} \in \mathbf{R}^{N \times C_{\text{out}} \times T \times V}$ is the result of the convolution operation, which involves $C_{\text{in}} \times C_{\text{out}} \times K_h \times K_w$ multiplication. This approach requires a large number of parameters and high computational cost, especially when the input data is large.

The proposed convolution operation is formulated as follows:

$$Y_{\text{dw}} = W_{\text{dw}} \times X \tag{15}$$

$$Y_{\text{final}} = W_{\text{pw}} \times Y_{\text{dw}} \tag{16}$$

where $W_{\text{dw}} \in \mathbf{R}^{C_{\text{in}} \times K_h \times K_w}$ is the depthwise filter applied to each input channel individually, $W_{\text{pw}} \in \mathbf{R}^{C_{\text{out}} \times C_{\text{in}} \times 1 \times 1}$ is the pointwise filter, which aggregates the output of the depthwise convolution across channels, $Y_{\text{dw}} \in \mathbf{R}^{N \times C_{\text{in}} \times T \times V}$ is the intermediate result after the depthwise convolution, and $Y_{\text{final}} \in \mathbf{R}^{N \times C_{\text{out}} \times T \times V}$ is the final output after the pointwise convolution.

3.2.2 Efficiency comparison

For TD-GCN[28] convolution operation, the computational complexity can be expressed as

$$\text{Cost}_{\text{TD-GCN}} = T \times V \times C_{\text{in}} \times C_{\text{out}} \times K_h \times K_w \tag{17}$$

This requires $C_{\text{in}} \times C_{\text{out}} \times K_h \times K_w$ multiplications for each spatial location in the input, resulting in a significant computational cost when C_{in} and C_{out} are large.

In contrast, the computational cost of depthwise separable convolution can be divided into depthwise and pointwise convolution. In depthwise convolution, each input channel is convolved independently with a single filter, leading to a computational cost,

$$\text{Cost}_{\text{dw}} = T \times V \times C_{\text{in}} \times K_h \times K_w \tag{18}$$

While in pointwise convolution, a 1×1 convolution is applied across all input channels, resulting in a computational cost,

$$\text{Cost}_{\text{pw}} = T \times V \times C_{\text{in}} \times C_{\text{out}} \tag{19}$$

As a result, the total cost of the depthwise separable convolution is

$$\text{Cost}_{\text{DSC}} = T \times V \times C_{\text{in}} \times K_h \times K_w + C_{\text{in}} \times C_{\text{out}} \tag{20}$$

Comparing the computational costs of the two approaches, we can see that the depthwise separable convolution is more efficient,

$$\frac{\text{Cost}_{\text{DSC}}}{\text{Cost}_{\text{TD-GCN}}} = \frac{C_{\text{in}} \times K_h \times K_w + C_{\text{in}} \times C_{\text{out}}}{C_{\text{in}} \times C_{\text{out}} \times K_h \times K_w} \tag{21}$$

For a large K_h and K_w, the depthwise separable convolution offers a substantial reduction in computation, especially when C_{in} and C_{out} are large.

3.3 Data augmentation

To enhance robustness against noise and occlusion in real-world environments, we propose a suite of augmentation techniques (see Fig. 3) to model conditions that lead to performance degradation. For instance, we recognise that an object occluding a camera view typically affects a continuous sequence of frames in real-world scenarios. This can occur randomly due to the cluttered real-world environments often seen. With these practical scenarios in mind, the following augmentation strategies are designed.

3.3.1 Random joint and frame occlusion

This method introduces variability into the spatial and temporal dimensions of the skeleton data, simulating potential occlusion situations encountered in real-world environments. It applies random erasure of joints and frames across randomly selected continuous sequences of frames. The process is controlled by the probability of erasure p and erasing sequences length range $L_{\min}$ and $L_{\max}$. First, frames are chosen for erasure based on probability p. Consecutive sequences of frames are then selected, with lengths between $L_{\min}$ and $L_{\max}$. For each selected sequence, random joints are set to zero to simulate joint occlusion, while for frame occlusion, all joint values are set to zero. After processing each sequence, a random number of frames are skipped before the subsequent process begins. The full procedure is outlined in Algorithm 1.

3.3.2 Skeleton rotation

This method introduces variability in spatial orientations to simulate different viewing angles. First, a random rotation vector is generated and transformed into a rotation matrix. The rotation matrix is then multiplied with the skeleton data to rotate the joint positions. The full algorithm is presented in Algorithm 2.

3.3.3 Jittering

The jittering method introduces random perturbations to skeleton data by adding Gaussian noise to the joint

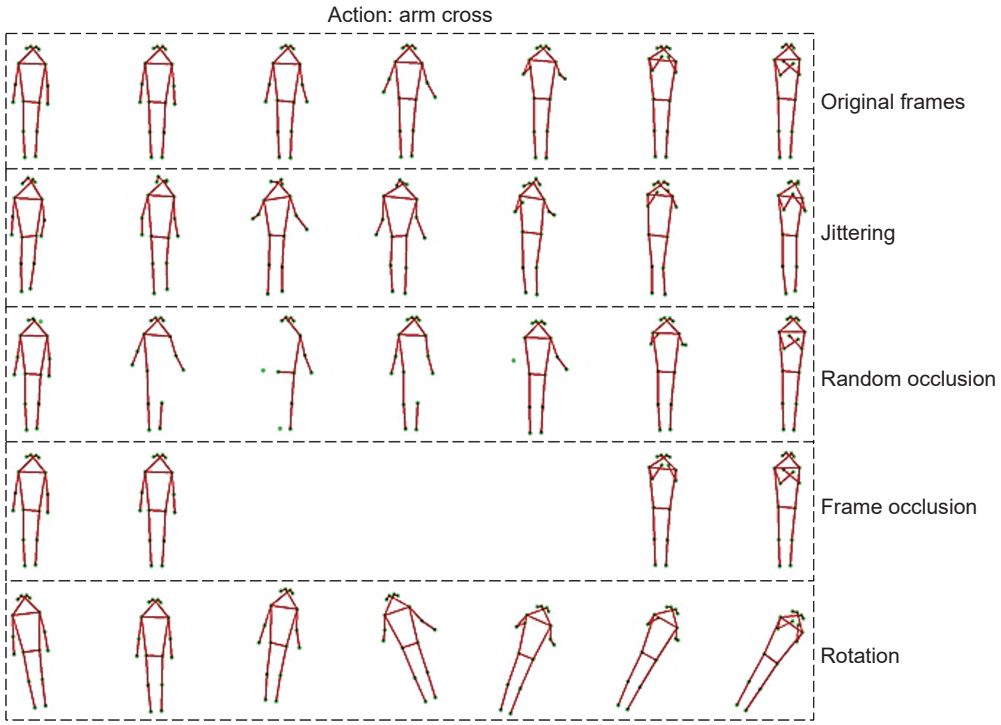


Fig. 3 Skeleton sample of "cross arm" action with data augmentation techniques: jittering, random occlusion, frame occlusion, and rotation.

position in selected frames. First, frames are chosen for noise addition based on a random decision controlled by the probability p_{frame}. For each selected frame, Gaussian noise is generated and added to the joint positions. The algorithm describing this process is detailed in Algorithm 3.

4 Experiment

In this section, we evaluate the accuracy, robustness, and efficiency of the proposed RE-TCN framework. We compare RE-TCN's performance to state-of-the-art skeleton-based action recognition methods and conduct comprehensive ablative studies.

4.1 Dataset

(1) NTU RGB+D 60[29]: The dataset comprises 56 880 skeleton sequences covering 60 actions performed by 40 subjects. Shahroudy et al.[29] suggested two primary evaluation benchmarks: **(a) Cross-view (X-view)**, where training data are captured from two camera angles, 0° (view 2) and 45° (view 3), while testing is conducted from −45° (view 1), and **(b) Cross-subject (X-sub)**, in which 20 subjects data are used for training while data from the remaining 20 subjects are reserved for testing. Following this approach, we report the top-1 recognition accuracy across both benchmarks.

(2) NorthWestern-UCLA (NW-UCLA)[30]: The NW-UCLA dataset includes 1494 sequences across 10 actions, captured from Kinect cameras, each positioned to provide different viewpoints. Following the evaluation protocol suggested by Ref. [30], data from the first two cameras are used for training, and the remaining camera's data are used for testing.

(3) SHREC'17[31]: The dataset comprises 2800 gesture sequences performs by 28 participants. Each of the 28 participants performs each gesture 10 times. The gestures are categorised into either 14 or 28 classes based on the gesture type. Consistent with the evaluation protocol outlined in Ref. [31], 1960 sequences are used for training, and 840 sequences are reserved for testing.

(4) DHG-14/28[32]: The dataset comprises 2800 gesture sequences performed 5 times each by 20 participants. As suggested by Ref. [32], the leave-one-subject-out cross-validation method are used for evaluation. This means that data from 19 participants are used for training and the remaining participant's data are reserved for testing. This evaluation process is repeated 20 times, and the final accuracy is reported as the average of these iterations.

Algorithm 1 Random joint and frame occlusion

Require: Original skeleton data $d \in \mathbf{R}^{C\times T\times V\times M}$

Ensure: Augmented skeleton data with occlusions $d_{\text{aug}} \in \mathbf{R}^{C\times T\times V\times M}$

Initialize d_{aug} with d;
$C, T, V, M \leftarrow$ Shape of d;
$p, L_{\min}, L_{\max} \leftarrow$ probability, min/max occlusion length;
for $m \leftarrow 0$ to $M-1$ **do**
  $t_{\text{current}} \leftarrow 0$;
  while $t_{\text{current}} < T$ **do**
    $t_{\text{remaining}} \leftarrow T - t_{\text{current}}$;
    if $t_{\text{remaining}} < L_{\min}$ **then**
    break;
    end if
    $L_{\text{occlusion}} \leftarrow$ Random int in $[L_{\min}, \min(L_{\max}, t_{\text{remaining}})]$
    if joint occlusion is selected **then**
      for $t \leftarrow t_{\text{current}}$ to $t_{\text{current}} + L_{\text{occlusion}} - 1$ **do**
        $J_{\text{occlude}} \leftarrow$ Random subset of joints from $\{1, 2, \ldots, V\}$;
        Set $d_{\text{aug}}[:, t, J_{\text{occlude}}, m] \leftarrow 0$;
      end for
    else if frame occlusion is selected **then**
      Set $d_{\text{aug}}[:, t_{\text{current}} : t_{\text{current}} + L_{\text{occlusion}} - 1, :, m] \leftarrow 0$;
    end if
    $t_{\text{skip}} \leftarrow$ Random int in $[1, \min(10, T - t_{\text{current}})]$;
    $t_{\text{current}} \leftarrow t_{\text{current}} + t_{\text{skip}}$;
  end while
end for

Algorithm 2 Skeleton rotation

Require: Skeleton data $d \in \mathbf{R}^{C\times T\times V\times M}$ and rotation angle θ
Ensure: Augmented skeleton data with random rotation $d_{\text{rot}} \in \mathbf{R}^{C\times T\times V\times M}$

Convert d to torch tensor d_{torch};
$C, T, V, M \leftarrow$ Shape of d_{torch};
Reshape and permute $d_{\text{torch}} \leftarrow$ Reshape to $(T, C, V \times M)$;
Initialize random rotation vector rot $\in \mathbf{R}^{T\times 3}$ with values in $[-\theta, \theta]$;
Apply a rotation function *_rot* to create a rotation matrix rot $\leftarrow$ _rot (rot) $\in \mathbf{R}^{T\times 3\times 3}$;
Multiply rotation matrix with skeleton data $d_{\text{torch}} \leftarrow$ rot $\times\, d_{\text{torch}}$;
Reshape and permute $d_{\text{torch}} \leftarrow$ Reshape back to (C, T, V, M);
Return augmented skeleton data d_{rot}

Algorithm 3 Jittering

Require: Skeleton data $d \in \mathbf{R}^{C\times T\times V\times M}$, Gaussian noise standard deviation σ, and frame selection probability p_{frame}

Ensure: Augmented skeleton data with jittering $d_{\text{jittered}} \in \mathbf{R}^{C\times T\times V\times M}$

Initialise augmented data $d_{\text{jittered}} \leftarrow$ copy of d;
$C, T, V, M \leftarrow$ Shape of d;
for $m \leftarrow 0$ to $M-1$ **do**
  for $t \leftarrow 0$ to $T-1$ **do**
    if rand ( ) $< p_{\text{frame}}$ **then**
      Generate Gaussian noise noise $\sim \mathcal{N}(0, \sigma^2)$ of shape $(C \times V)$;
      Add noise to the selected frame: $d_{\text{jittered}}[:, t, :, m] \leftarrow d_{\text{jittered}}[:, t, :, m] + \text{noise}$;
    end if
  end for
end for
Return augmented skeleton data d_{jittered}

4.2 Implementation details

The model is trained using Stochastic Gradient Descent (SGD) with a warm-up strategy. The learning rate is initialised at 0.1, and a momentum of 0.9 is applied. Model checkpointing is used to save the model based on optimal performance on the validation set. For the SHREC'17 and DHG-14/28 datasets, a batch size of 32 is used, with a weight decay of 0.0001 and a learning rate decay factor of 0.1. The NTU RGB+D 60 dataset is trained with a batch size of 64, a weight decay of 0.0004 and a learning rate decay factor of 0.1. For the NW-UCLA dataset, the model is trained with a batch size of 16, a weight decay of 0.0001 and a learning rate decay factor of 0.1. Based upon findings from ablation studies, the following parameter settings are used for the ATW mechanism: the reduction ratio is set to 8, the mean pooling strategy is used, placement in the model architecture is set to late, and two layers with a reduction ratio are implemented.

4.3 Ablation study and parameter tunning

To analyse the various components of the proposed method, we perform extensive experiments using the NW-UCLA dataset as a case study.

4.3.1 ATW and DSC

We evaluate the effectiveness of the proposed ATW and DSC in enhancing model accuracy and optimising parameter count, as summarised in Table 1. For this evaluation, baseline training is conducted using the TD-GCN[28] and TD-GDSCN[18] model, and the same training parameter settings are maintained across all experiments. The only modifications involve the integration of different components of the proposed method. As shown in Table 1, the integration of DSC improves both accuracy and parameter efficiency. Furthermore, the addition of ATW yields an additional

Table 1 Impact of ATW and DSC on accuracy. The best results are highlighted in bold.

Method	Number of parameters ($\times 10^6$)	Accuracy (%)
TD-GCN	1.35	94.40
TD-GDSCN	**1.23**	95.05
RE-TCN + DSC	**1.23**	95.69
RE-TCN + DSC + ATW	1.24	**96.34**

increase in accuracy, although with a minimal parameter increase of 0.01. These results validate the contribution of DSC and ATW in enhancing the baseline model's performance.

4.3.2 Reduction ratio

To enable efficient convolution operations, one design choice in ATW is to reduce the number of channels by a factor of n in the first convolution operation. We conduct an experiment to select an appropriate reduction ratio. Initially, we use a reduction ratio of 8 to train the model as a baseline. We then test different reduction ratios and present the results in Table 2. The results show that higher reduction ratios lead to fewer parameter counts, with a ratio of 64 yielding the fewest parameters. However, accuracy fluctuats among the different ratios, with the ratio of 8 achieving the highest accuracy. Since the increase in parameter count with a ratio of 8 compared to 64 is minimal, and the accuracy improvement is significant, we use the reduction ratio of 8 in subsequent experiments.

4.3.3 Pooling strategy

To compute the attention weights along the temporal dimension, the ATW mechanism first collapses the joint dimension. This reduces the spatial complexity and focuses on the temporal aspect of the features. We conduct an experiment to determine the most appropriate pooling mechanism for collapsing the joint dimension. We test various pooling strategies and present the results in Table 3. The results show that all pooling strategies have the same effect on parameter count. However, the mean strategy's accuracy is significantly higher compared to others. Therefore, we select it as the pooling strategy for subsequent experiments.

Table 2 Impact of reduction ration on accuracy. The best results are highlighted in bold.

Reduction ratio	Accuracy (%)	Number of parameters
4	93.32	1 258 896
8	**96.34**	1 242 480
16	92.24	1 234 272
32	94.40	1 230 168
64	93.75	**1 228 116**

Table 3 Impact of pooling strategy on accuracy. The best results are highlighted in bold.

Pooling strategy	Accuracy (%)	Number of parameters ($\times 10^6$)
Max	93.10	1.24
Adaptive	94.18	1.24
Mean	**96.34**	1.24
Global max	91.81	1.24
Global mean	83.41	1.24

4.3.4 ATW location in model architecture

We conduct an experiment to identify the optimal placement location of the ATW mechanism within the overall model architecture. Various positions are tested, with the result presented in Table 4. The results indicate that the placement of ATW affects both accuracy and parameter count. Introducing the ATW mechanism earlier in the network minimises parameter count, though this configuration does not achieve the highest accuracy. Conversely, positioning ATW later in the network produces the highest accuracy, with only a minimal increase in parameter count relative to early placement.

4.3.5 Efficient design strategy

One key design choice in ATW mechanism is to use a two-step convolution operation with an intermediate dimensionality reduction. We conduct an experiment using various combinations of convolution layers to assess their impact on accuracy and parameter count. The result is presented in Table 5. We train and evaluate two sets of models: one with only full convolution layers (without a dimensionality reduction), and another with a dimensionality reduction layer. In both sets, we observe that the parameter count increases with the number of convolution layers.

Table 4 Impact of ATW location in model architecture on accuracy. The best results are highlighted in bold.

Location in model architecture	Accuracy (%)	Number of parameters
Early	93.10	**1 226 904**
Middle	92.75	1 230 048
Late	**96.34**	1 242 480
Early + Middle	92.89	1 231 144
Late + Middle	92.67	1 246 720
Early + Middle + Late	93.10	1 247 816

Table 5 Impact of design strategy on accuracy and parameter count, where layer is the convolution layer, and W-red stands for with reduction ration of 8. The best results are highlighted in bold.

Design strategy	Accuracy (%)	Number of parameters
One layer	93.97	1 291 600
Two layers	94.40	1 357 392
Three layers	81.41	1 423 184
Four layers	88.58	1 488 976
Two layers/W-red	**96.34**	**1 242 480**
Three layers/W-red	89.44	1 308 272
Four layers/W-red	91.51	1 374 064

However, accuracy does not consistently improve with more layers. In both sets, the accuracy is higher with two convolution layers, with the dimensionality reduction model performing best. Based on this finding, we adopt this setup for subsequent experiments.

4.3.6 DAS

We explore the accuracy of different data augmentation strategies to evaluate their effect. The result is presented in Table 6. First, we train the model without data augmentation and test it on data with jittering and occlusion. Next, we introduce various augmentation types and test them again on data with jittering and occlusion. The results show that accuracy increases significantly with the introduction of data augmentation. This validates the effectiveness of data augmentation in ensuring robust action recognition against noise.

Table 6 Influence of augmentation types, where "N" is noise (jittering and occlusion), and R stands for skeleton rotation. The best results are highlighted in bold.

Augmentation type	Jittering (%)	Occlusion (%)
RE-TCN	88.36	82.54
RE-TCN+R	92.24	90.73
RE-TCN+R+N	**94.18**	**94.40**

4.4 Performance comparison with TD-GCN

4.4.1 Comparison of accuracy and parameter count

We conduct an experiment using the SHREC'17 and NW-UCLA datasets to compare the accuracy and parameter count of RE-TCN and TD-GCN. For a fair comparison, we use the same training parameter settings and joint data modality for both models. The result is presented in Table 7. RT-TCN outperforms TD-GCN across both datasets, with accuracy improvements of 1.54% for NW-UCLA, 3.54% for SHREC'17 14 gesture, and 6.38% for SHREC'17 28 gesture. In terms of computational efficiency, RE-TCN reduces the parameter count by 0.11% for both datasets. These findings demonstrate that RE-TCN effectively enhances both computational efficiency and accuracy.

4.4.2 Performance comparison on cross-subject evaluation

We conduct an experiment utilising the DHG-14/28 dataset to compare the generalisability capability of TD-GCN and RE-TCN. We aim to evaluate the ability of the model to generalise across new participants whose data are not used to train the model. We utilise the DHG-14/28 dataset as it allows effective cross-subject evaluation through the leave-one-subject-out cross-validation method suggested by Ref. [32]. To ensure a fair comparison, we employ the same training parameter settings and joint data modality for both models. The result is presented in Table 8.

Table 7 Comparison of accuracy and parameter count. The best results are highlighted in bold.

Method	Dataset	Number of parameters ($\times 10^6$)	Accuracy (%)
TD-GCN	NW-UCL	1.35	94.8
RE-TCN (Ours)	NW-UCL	**1.24**	**96.34**
TD-GCN	SHREC'17 14	1.36	96.31
RE-TCN (Ours)	SHREC'17 14	**1.25**	**99.85**
TD-GCN	SHREC'17 28	1.36	93.57
RE-TCN (Ours)	SHREC'17 28	**1.25**	**99.95**

Table 8 Accuracy recognition per subjects for the DHG-14/28 dataset. "–" denotes results not provided by TD-GCN. The best results are highlighted in bold.

(%)

Dataset type	Subject																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
14 gestures (RE-TCN)	88.57	80.00	97.86	**95.00**	**92.86**	87.86	87.14	92.86	91.43	**95.00**	95.00	**92.14**	88.57	**90.00**	95.71	91.14	**87.14**	91.43	97.86	88.57
14 gestures (TD-GCN)	**90.00**	–	**98.57**	93.57	90.00	–	–	92.86	91.43	92.86	**95.71**	90.00	–	88.57	**97.14**	**96.43**	86.43	–	97.86	–
28 gestures (RE-TCN)	**90.00**	72.86	92.14	88.57	**90.71**	84.29	91.43	86.43	**85.71**	92.86	**92.86**	90.71	83.57	84.29	95.00	**90.71**	84.29	90.71	93.57	83.57
28 gestures TD-GCN)	89.29	–	**93.57**	88.57	85.00	–	–	**92.14**	83.57	92.86	89.29	**92.14**	–	**88.57**	**95.71**	87.86	**86.43**	–	**94.29**	–

From the result, we observe that performance across the subjects varies, highlighting the differences that exist when individuals perform the same action. These variations in how actions are performed by different people make recognising actions more challenging for certain individuals, as evidenced by the varying accuracy amongst subjects. Nevertheless, despite this challenge, the overall accuracy of RE-TCN across all the subjects for both 14 and 28 action types is notably high. When compared to TD-GCN, the accuracy is comparable, with RE-TCN demonstrating superior performance in some subjects while achieving equal accuracy in others. Overall, the result underscores the capability of RE-TCN to generalise effectively across different subjects.

4.5 Comparison with state-of-the-art methods

Here, we compare RE-TCN's accuracy and robustness against noise and occlusion with state-of-the-art methods using the SHREC'17, NW-UCLA, and NTU-RGB+D 60 datasets. Details of the comparison are discussed below.

4.5.1 Accuracy comparison with state-of-the-art methods

We compare RE-TCN with the state-of-the-art methods on the SHREC'17 and NW-UCLA datasets, which are skeleton-based gesture and skeleton-based action recognition datasets. Some methods use an ensemble approach, fusing results from joint, bone, and motion modalities, while others use only joint data. For a fair comparison, we report accuracy using only joint data modality. The results are presented in Tables 9 and 10.

On both datasets, our model (RE-TCN) outperforms existing methods. In Table 9, using 14 and 28 gesture classes, classification accuracies are 99.95% and 99.85%, respectively. This surpasses the current best-performing method in Ref. [52] by 2.85% for 14 gestures and that in Ref. [58] by 5.05% for 28 gestures. In Table 10, the classification accuracy is 96.34%, outperforming the current best-performing methods[54, 66] by 1.04%. These results demonstrate the effectiveness of the proposed in enhancing accuracy for both gestures and action recognition.

Table 9 Accuracy comparison with state-of-the-art methods using SHREC'17 dataset. The best results are highlighted in bold.

(%)

Method	14 gestures	28 gestures
ST-GCN[40]	92.70	87.70
MS-ISTGCN[58]	96.70	94.90
ST-TS-HGR-Net[59]	94.30	89.40
HPEV[60]	94.90	92.30
DSTA-Net[52]	97.00	93.90
TD-GCN[28]	96.31	93.57
RE-TCN (Ours)	**99.85**	**99.95**

Table 10 Accuracy comparison with state-of-the-art methods using NW-UCLA dataset. The best results are highlighted in bold.

Method	Accuracy (%)
Lie Group[60]	74.20
Actionlet ensemble[61]	76.00
HBRNN-L[62]	78.50
Skeleton Visualisation[63]	86.10
Ensemble TS-LSTM[64]	89.20
AGC-LSTM[43]	93.30
Shift-GCN[25]	94.60
DC-GCN+ADG[66]	95.30
FGCN[54]	95.30
TD-GCN[28]	94.80
RE-TCN (Ours)	**96.34**

4.5.2 Robustness to occlusion

To evaluate the robustness of the proposed method against occlusion, we test RE-TCN with occluded data from various perspectives. Our tests cover both spatial and temporal occlusions, reflecting scenarios likely to be encountered in real-world ambient-assisted living environments. For a fair comparison, we compare the performance of RE-TCN with state-of-the-art methods designed to handle occlusions. We design our occlusion experiments following the same conditions described in Refs. [25, 57, 67] and compare our results with their methods. We use the NTU-RGB+D 60 X-sub dataset and define three types of occlusions: frame, body part, and random occlusion, as described in Refs. [25, 57, 67]. The details of these occlusion types and the comparison results are discussed below.

(1) Frame occlusion

This occlusion type simulates temporal occlusion. As described in Refs. [25, 67], we randomly occlude consecutive sequences of frames from an action sequence. We set the length of the occluded consecutive frames to 10, 20, 30, 40, and 50. The experimental result is shown in Table 11. We conduct two experiments: one with a low probability and another with a high probability of frame occlusion occurring. The results reveal that methods not designed

Table 11 Experiment results (accuracy) with frame occlusion on NTU-RGB+D 60 X-sub benchmark. The best results are highlighted in bold.

(%)

Method	Number of occluded frames					
	0	10	20	30	40	50
ST-GCN[40]	80.7	69.3	57.0	44.5	34.5	24.0
SR-TSL[68]	84.8	70.9	62.6	48.8	41.3	28.8
STIGCN[69]	88.8	70.4	51.0	38.7	23.8	8.0
MS-G3D[70]	87.3	77.6	65.7	54.3	41.9	30.1
CTR-GCN[50]	87.5	72.4	54.1	35.6	22.4	11.5
TCA-GCN[71]	**90.2**	84.4	74.6	58.1	42.3	25.6
HD-GCN[72]	86.8	57.0	29.5	18.5	11.2	7.04
2s-AGCN[41]	88.5	74.8	60.8	49.7	38.2	28.0
1s RA-GCN[67]	85.8	81.6	72.9	61.6	47.9	34.0
2s RA-GCN[67]	86.7	83.0	76.4	65.6	53.1	39.5
3s RA-GCN[67]	87.3	83.9	76.4	66.3	53.2	38.5
1s PDGCN[25]	85.7	81.9	75.4	66.4	54.9	40.0
2s PDGCN[25]	87.4	83.8	76.7	66.8	55.1	40.6
3s PDGCN[25]	87.5	83.9	76.6	66.7	53.9	40.0
RE-TCN (P = 0.50) (ours)	89.33	87.68	86.75	84.18	85.13	81.24
RE-TCN (P = 0.01) (ours)	89.85	**89.77**	**89.75**	**89.68**	**89.67**	**89.60**

for occlusion robustness perform poorly as the occlusion length increases. Even methods designed to be robust against occlusion, such as Refs. [25, 67], show a significant performance degradation with increased occlusion length. In contrast, our method demonstrates minimal performance degradation as occlusion length increases. Moreover, our results outperform existing methods across all the occlusion lengths, for both low and high probability scenarios. The performance gap between our method and those of Refs. [25, 67] widens as the occlusion length increases, reaching about 49% and 50.1%, respectively, when the occlusion length is 50.

(2) Body part occlusion

Body part occlusion aims to simulate scenarios where some body parts of a person are occluded by objects or self-occlusion, which is common in real-world unstructured environments. As defined in Refs. [25, 67], we occlude the left arm, right arm, two hands, two legs, and torso when testing the model. The experiment result is presented in Table 12. The effects of occluding different body parts vary, with some parts showing better performance than others. Similar to frame occlusion, methods not designed to be robust against occlusion suffer performance degradation when body parts are occluded. For Refs. [25, 67], the performance notably improves with body part occlusion. In comparison, our method outperforms existing methods across all body parts by a significant margin, validating the robustness of our method against occlusion.

(3) Random occlusion

Random occlusion simulates how skeleton data might be obscured in real-world environments. Following the approach in Ref. [67], we set the occlusion probabilities to 0.2, 0.3, 0.4, 0.5, and 0.6 with the result shown in Table 13. We also compare our results with the approach in Ref. [57], using their

Table 12 Experiment results (accuracy) with body part occlusion on NTU-RGB+D 60 X-sub benchmark. The best results are highlighted in bold.

(%)

Method	Occlusion body parts					
	None	Left arm	Right arm	Two hands	Two legs	Trunk
ST-GCN[40]	80.7	71.4	60.5	62.6	77.4	50.2
SR-TSL[68]	84.8	70.6	54.3	48.6	74.3	56.2
STIGCN[69]	88.8	12.7	11.5	18.3	45.5	20.9
MS-G3D[70]	87.3	31.3	23.8	17.1	78.3	61.6
CTR-GCN[50]	87.5	13.0	12.5	12.7	21.0	36.3
TCA-GCN[71]	**90.2**	75.4	53.4	70.8	75.2	78.6
HD-GCN[72]	86.7	67.1	55.7	56.7	74.8	61.3
2s-AGCN[41]	88.5	72.4	55.8	82.1	74.1	71.9
1s RA-GCN[67]	85.8	69.9	54.0	66.8	82.4	64.9
2s RA-GCN[67]	86.7	75.9	62.1	69.2	83.3	72.8
3s RA-GCN[67]	87.3	74.5	59.4	74.2	83.2	72.3
1s PDGCN[25]	85.7	73.4	60.4	65.9	83.0	71.2
2s PDGCN[25]	87.4	76.4	62.0	74.4	84.8	70.4
3s PDGCN[25]	87.5	76.0	62.0	75.4	85.0	74.3
RE-TCN (ours)	89.78	**89.11**	**88.60**	**88.26**	**89.46**	**89.29**

Table 13 Experiment results (accuracy) with random occlusion on NTU-RGB+D 60 X-sub benchmark. The best results are highlighted in bold.

(%)

Method	Occlusion probability					
	0	0.2	0.3	0.4	0.5	0.6
ST-GCN[40]	80.7	12.4	6.6	6.2	4.0	4.2
SR-TSL[68]	84.8	43.0	25.2	12.1	6.0	3.7
2s-AGCN[41]	88.5	38.5	22.8	13.4	8.5	6.1
RA-GCN[56]	85.9	84.1	81.7	77.2	70.0	57.4
1s RA-GCN[67]	80.0	75.1	68.4	57.4	44.7	27.6
2s RA-GCN[67]	82.5	79.7	76.2	71.0	62.0	48.7
3s RA-GCN[67]	82.7	79.8	75.6	68.9	58.1	43.7
RE-TCN (ours)	**88.7**	**88.6**	**88.4**	**88.4**	**88.6**	**88.2**

occlusion probabilities of 0.08, 0.10, 0.12, and 0.15, as presented in Table 14.

Table 13 reveals a significant performance drop as occlusion probabilities increase for those methods not designed to handle occlusion. While Ref. [67] shows improved robustness, the method still suffers noticeable degradation at higher occlusion levels. Our method, however, experiences only a minimal performance loss as the probability increases, and outperforms existing methods across all occlusion probabilities.

Table 14 shows results for a denoising method used as a preprocessing module with state-of-the-art approaches. While this method maintains stable performance across various occlusion probabilities, our approach consistently outperforms it at all levels.

The superior performance of our method as demonstrated in both Tables 13 and 14, underscores its robustness against occlusion.

Table 14 Experiment results (accuracy) with random occlusion on NTU-RGB+D 60 X-sub benchmark: The symbol "w" denotes the use of DAE denoising method[57]. The best results are highlighted in bold.

(%)

Method	Occlusion probability				
	0	0.08	0.10	0.12	0.15
EfficientGCN[55] w[57]	87.74	87.66	87.66	87.57	87.51
ST-GCN++[73] w[57]	87.80	87.47	87.37	87.38	87.23
CTR-GCN[50] w[57]	89.20	89.19	89.19	89.12	88.12
AAGCN[74] w[57]	88.71	88.42	88.42	88.37	88.42
MS-G3D[70] w[57]	88.75	88.71	88.68	88.67	88.57
RE-GCN (ours)	**89.78**	**89.35**	**89.29**	**89.20**	**88.92**

4.5.3 Robustness to noise

This is designed to simulate the effect of noise in skeleton data, a common challenge in real-world environments. Following the approach described in Ref. [67], we design two experiments with different Gaussian noises as shown in Tables 15 and 16. We set the probability for every joint to 0.02, 0.04, 0.06, 0.08, and 0.10. Additionally, we adopt the approach from Ref. [57] and set the jittering probability to 0.05, 0.1, 0.2 and 0.3, as shown in Table 17.

In Tables 15 and 16, we observe a significant performance degradation as the jittering probability increases, even with the method in Ref. [67] designed to handle jittering in skeleton data. By contrast, our method demonstrates consistent performance across all the different probability levels, outperforming existing methods. Notably, for both tested values of σ, the performance gap between our method and the one in Ref. [67] widens significantly as the jittering probability increases with a margin of 52.96% and

Table 15 Experiment results (accuracy) with jittering skeletons ($\mu = 0$ and $\sigma = 0.1$) on NTU-RGB+D 60 X-sub benchmark. The best results are highlighted in bold.

(%)

Method	Jittering probability					
	0	0.02	0.04	0.06	0.08	0.10
ST-GCN[40]	80.7	66.4	44.1	32.7	13.3	7.0
SR-TSL[68]	84.8	70.4	53.2	41.0	33.9	21.4
2s-AGCN[41]	88.5	74.9	60.9	41.9	29.4	20.6
RA-GCN[56]	85.9	73.2	59.8	45.3	41.6	34.5
1s RA-GCN[67]	85.8	84.1	66.1	34.2	22.2	13.9
2s RA-GCN[67]	86.7	70.0	55.3	48.2	41.5	36.4
3s RA-GCN[67]	87.3	84.2	72.4	61.6	42.4	28.7
RE-TCN (ours)	**89.58**	**89.47**	**89.49**	**89.45**	**89.22**	**89.36**

Table 16 Experiment results (accuracy) with jittering skeletons ($\mu = 0$ and $\sigma = 0.05$) on NTU-RGB+D 60 X-sub benchmark. The best results are highlighted in bold.

(%)

Method	Jittering probability					
	0	0.02	0.04	0.06	0.08	0.10
ST-GCN[40]	80.7	76.4	65.1	50.2	32.8	19.5
SR-TSL[68]	84.8	69.4	55.3	50.1	46.6	39.2
2s-AGCN[41]	88.5	78.9	79.8	76.8	72.6	60.7
RA-GCN[56]	85.9	83.8	81.3	75.3	69.2	61.4
1s RA-GCN[67]	85.8	82.4	77.1	72.3	63.8	49.9
2s RA-GCN[67]	86.7	83.8	77.3	71.6	61.6	58.5
3s RA-GCN[67]	87.3	87.0	84.5	81.1	72.9	61.4
RE-TCN (ours)	**89.12**	**89.04**	**89.01**	**88.99**	**88.90**	**88.97**

Table 17 Experiment results (accuracy) with random jittering on NTU-RGB+D 60 X-sub benchmark: The symble "w" denotes the use of DAE denoising method[57]. The best results are highlighted in bold.

(%)

Method	Jittering probability				
	0	0.05	0.10	0.20	0.30
EfficientGCN[55] w[57]	87.62	87.57	87.59	87.60	87.58
ST-GCN++[73] w[57]	87.89	87.80	87.75	87.58	87.31
CTR-GCN[50] w[57]	89.07	89.11	89.03	89.09	**88.76**
AAGCN[74] w[57]	88.57	88.59	88.65	88.65	88.45
MS-G3D[70] w[57]	88.60	88.64	88.62	88.61	88.34
RE-GCN (ours)	**89.58**	**89.56**	**89.36**	**88.97**	88.41

27.57% for $\sigma = 0.1$ and $\sigma = 0.05$, respectively, at a probability of 0.1.

In Table 17, while the performance of the method in Ref. [57] demonstrates a stable performance across probability levels, our approach outperforms it across all the probability levels.

Overall, the results in Tables 15–17 highlight the robustness of our approach in mitigating the effects of jittering.

5 Per-Class Classification Performance

We evaluate the classification performance for each class using NTU RGB+D 60, Northwestern-UCLA, SHREC'17, and DHG-14/28 datasets. For this evaluation, we compute the confusion matrix, as presented in Fig. 4 and classification report, which is provided in the Electronic Supplementary Material (ESM) of the online version of this article. As shown in the confusion matrix, the model achieves high accuracy across all classes, with the exception of the DHG-14/28 dataset, where the model occasionally misclassifies the grab class as the pinch class. This confusion is reasonable given the high similarity between the two classes. Apart from this, the model consistently demonstrates high accuracy across all classes in the other datasets. This class-wise classification performance highlights the model's robustness and its ability to effectively handle diverse range of classes.

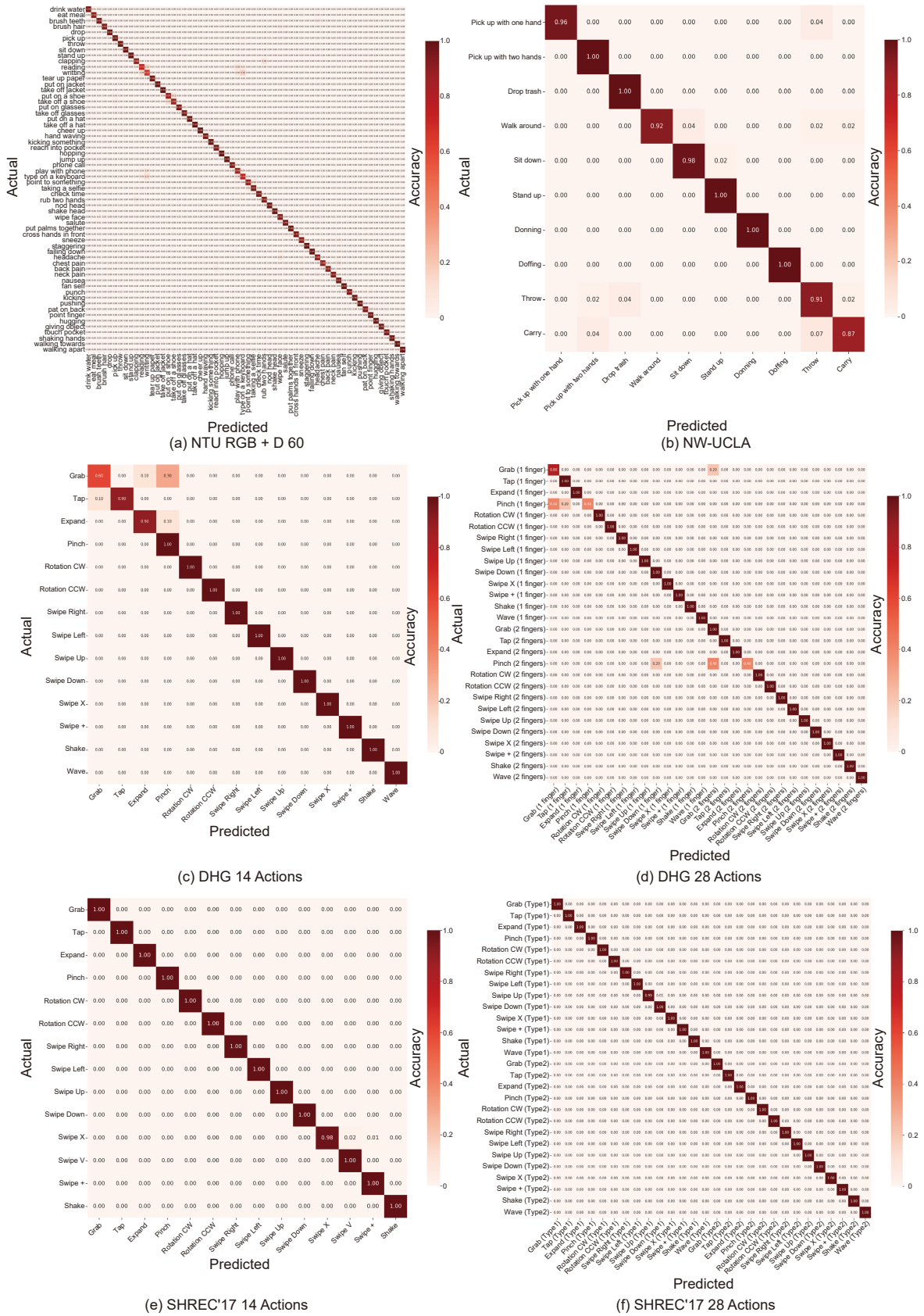
6 Real-Time Application for Human Action Recognition

We develop a human action recognition system using RE-TCN to demonstrate its practical use in real-world, unstructured environments. The system is tested in a challenging environmental condition where objects partially block the camera's view, resulting in noisy data and occlusions. Using Mediapipe Pose[75], the system processes RGB image sequences to estimate human poses, which then feeds into the model for real-time action recognition. We test the system on a standard PC with an Intel Core i5 processor running Ubuntu 20.04 LTS, without a dedicated GPU. As shown in Fig. 5, the system successfully identifies actions in real time. The fast inference speed demonstrates that the model can perform effectively even on devices with limited resources. The model's reliable performance despite noise and occlusion demonstrates its suitability for practical applications. These results indicate strong potential for real-world deployment for ambient assisted living applications.

7 Conclusion and Future Study

In this paper, we present RE-TCN, a model designed to address the challenges of noisy data, occlusion and computational cost in real-world ambient assisted living environments. The proposed RE-TCN model incorporates three key components: ATW, DSC, and data augmentation techniques. Through extensive experiments on the NTU-RGB+D 60, NW-UCLA, and SHREC'17 datasets, we demonstrate the effectiveness and robustness of the model. To evaluate the robustness of RE-TCN in the presence of noise and occlusions, we conduct tests simulating various real-world conditions. Across all test configurations, RE-TCN consistently achieves state-of-the-art performance. Additionally, the model exhibits superior accuracy and computational efficiency compared to existing approaches, indicating its potential for facilitating accurate action recognition in real-world ambient assisted living environments.

Although the model achieves state-of-the-art performance on different testing configurations, there remain certain limitations that, if addressed, could further strengthen the model's applicability in practice. First, while RE-TCN demonstrates generalisability comparable to the current methods, there is a scope to optimise the architecture to improve its ability to handle data from individuals not represented during training. Secondly, although the datasets employed are comprehensive, using a dataset specifically collected from care homes or similar settings would more directly ensure that the model is applicable to its intended context. Thirdly, although the experiments accounted for various challenges likely to be



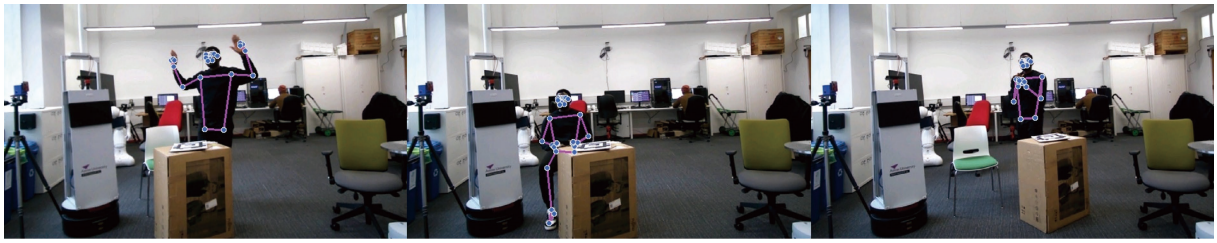
(a) NTU RGB + D 60 (b) NW-UCLA

(c) DHG 14 Actions (d) DHG 28 Actions

(e) SHREC'17 14 Actions (f) SHREC'17 28 Actions

Fig. 4 Confusion matrices showing classification performance for each class in the NTU RGB+D 60, Northwestern-UCLA, SHREC'17, and DHG-14/28 datasets.

(a)



(b)

Fig. 5 Real time action recognition: (a) action recognition without occlusions and (b) action recognition with occlusions. Predicted action: The action is recognised by the model. Inference time: The time taken for the model to generate a prediction. Inference + processing time: The total processing time that spans from frame capture, pose extraction, and model prediction.

encountered in real-world conditions, the model has yet to be deployed in a living setting. Field evaluations, such as monitoring daily activities in care homes or the residences of elderly individuals, could yield invaluable insights into its practical utility.

Addressing these limitations will not only enhance RE-TCN adaptability and scalability, but also further emphasise its relevance across a range of real-world environments.

Electronic Supplementary Material

Supplementary materials of the classification performance for each class using NTU RGB+D 60, Northwestern-UCLA, SHREC'17, and DHG-14/28 datasets are available in the online version of this article at http://doi.org/10.26599/BDWA.2015.9020003.

Acknowledgment

The model was trained on the Aston EPS Machine Learning Server, funded by the EPSRC Core Equipment Fund (No. EP/V036106/1).

References

[1] United Nations Department of Economic and Social Affairs, *World Population Prospects 2019*: *Highlights*. New York NY, USA: United Nations, 2019.

[2] C. H. Jones and M. Dolsten, Healthcare on the brink: Navigating the challenges of an aging society in the united states, *npj Aging*, vol. 10, no. 1, p. 22, 2024.

[3] C. Bosch-Farré, M. C. Malagón-Aguilera, D. Ballester-Ferrando, C. Bertran-Noguer, A. Bonmatí-Tomàs, S. Gelabert-Vilella, and D. Juvinyà-Canal, Healthy ageing in place: Enablers and barriers from the perspective of the elderly. A qualitative study, *Int. J. Environ. Res. Public Health*, vol. 17, no. 18, p. 6451, 2020.

[4] M. Lette, A. Stoop, E. Gadsby, E. A. Ambugo, N. C. Mateu, J. Reynolds, G. Nijpels, C. Baan, and S. R. de Bruin, Supporting older people to live safely at home – findings from thirteen case studies on integrated care across Europe, *Int. J. Integr. Care*, vol. 20, no. 4, p. 1, 2020.

[5] S. Blackman, C. Matlo, C. Bobrovitskiy, A. Waldoch, M. L. Fang, P. Jackson, A. Mihailidis, L. Nygård, A. Astell, and A. Sixsmith, Ambient assisted living technologies for aging well: A scoping review, *J. Intell. Syst.*, vol. 25, no. 1, pp. 55–69, 2016.

[6] M. Bennasar, B. A. Price, D. Gooch, A. K. Bandara, and B. Nuseibeh, Significant features for human activity recognition using tri-axial accelerometers, *Sensors*, vol. 22, no. 19, p. 7482, 2022.

[7] C. M. Ranieri, S. MacLeod, M. Dragone, P. A. Vargas, and R. A. F. Romero, Activity recognition for ambient assisted living with videos, inertial units and ambient sensors, *Sensors*, vol. 21, no. 3, p. 768, 2021.

[8] V. Dentamaro, V. Gattulli, D. Impedovo, and F. Manca,

Human activity recognition with smartphone-integrated sensors: A survey, *Expert Syst. Appl.*, vol. 246, p. 123143, 2024.

[9] Khimraj, P. K. Shukla, A. Vijayvargiya, and R. Kumar, Human activity recognition using accelerometer and gyroscope data from smartphones, in *Proc. 2020 Int. Conf. Emerging Trends in Communication, Control and Computing*, Lakshmangarh, India, 2020, pp. 1–6.

[10] S. Cristina, V. Despotovic, R. Pérez-Rodríguez, and S. Aleksic, Audio- and video-based human activity recognition systems in healthcare, *IEEE Access*, vol. 12, pp. 8230–8245, 2024.

[11] D. K. Basuki, R. Z. Fhamy, M. I. Awal, L. H. Iksan, S. Sukaridhoto, and K. Wada, Audio based action recognition for monitoring elderly dementia patients, in *Proc. 2022 Int. Electronics Symp.*, Surabaya, Indonesia, 2022, pp. 522–529.

[12] G. Bhola and D. K. Vishwakarma, A review of vision-based indoor HAR: State-of-the-art, challenges, and future prospects, *Multimed. Tools Appl.*, vol. 83, no. 1, pp. 1965–2005, 2024.

[13] C. Wang and J. Yan, A comprehensive survey of RGB-based and skeleton-based human action recognition, *IEEE Access*, vol. 11, pp. 53880–53898, 2023.

[14] L. Lo Presti and M. La Cascia, 3D skeleton-based human action classification: A survey, *Pattern Recognit.*, vol. 53, pp. 130–147, 2016.

[15] M. Fanuel, X. Yuan, H. N. Kim, L. Qingge, and K. Roy, A survey on skeleton-based activity recognition using graph convolutional networks (GCN), in *Proc. 2021 12th Int. Symp. on Image and Signal Processing and Analysis*, Zagreb, Croatia, 2021, pp. 177–182.

[16] A. Jain, R. Akerkar, and A. Srivastava, Privacy-preserving human activity recognition system for assisted living environments, *IEEE Trans. Artif. Intell.*, vol. 5, no. 5, pp. 2342–2357, 2024.

[17] C. K. Htoo and M. M. Sein, Privacy preserving human fall recognition using human skeleton data, in *Proc. 2023 IEEE Conf. Computer Applications*, Yangon, Myanmar, 2023, pp. 276–281.

[18] V. G. Zakka, Z. Dai, and L. J. Manso, Action recognition for privacy-preserving ambient assisted living, in *Proc. 1st Int. Conf. Artificial Intelligence in Healthcare*, Swansea, UK, 2024, pp. 203–217.

[19] W. Guo, Y. Du, X. Shen, V. Lepetit, X. Alameda-Pineda, and F. Moreno-Noguer, Back to MLP: A simple baseline for human motion prediction, in *Proc. 2023 IEEE/CVF Winter Conf. Applications of Computer Vision*, Waikoloa, HI, USA, 2023, pp. 4798–4808.

[20] Y. Jiang, X. Yang, J. Liu, and J. Zhang, A lightweight hierarchical model with frame-level joints adaptive graph convolution for skeleton-based action recognition, *Secur. Commun. Networks*, vol. 2021, no. 1, p. 2290304, 2021.

[21] B. Zhang, J. Han, Z. Huang, J. Yang, and X. Zeng, A real-time and hardware-efficient processor for skeleton-based action recognition with lightweight convolutional neural network, *IEEE Trans. Circuits Syst. II*: *Express Briefs*, vol. 66, no. 12, pp. 2052–2056, 2019.

[22] Z. Chen, H. Wang, and J. Gui, Occluded skeleton-based human action recognition with dual inhibition training, in *Proc. 31st ACM Int. Conf. Multimedia*, Ottawa, Canada, 2023, pp. 2625–2634.

[23] J. Sanchez, C. Neff, and H. Tabkhi, Real-world graph convolution networks (RW-GCNs) for action recognition in smart video surveillance, in *Proc. 2021 IEEE/ACM Symp. on Edge Computing*, San Jose, CA, USA, 2021, pp. 121–134.

[24] Y. Yoon, J. Yu, and M. Jeon, Predictively encoded graph convolutional network for noise-robust skeleton-based action recognition, *Appl. Intell.*, vol. 52, no. 3, pp. 2317–2331, 2022.

[25] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, Skeleton-based action recognition with shift graph convolutional network, in *Proc. 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, pp. 180–189.

[26] Y. F. Song, Z. Zhang, C. Shan, and L. Wang, Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition, in *Proc. 28th ACM Int. Conf. Multimedia*, Seattle, WA, USA, 2020, pp. 1625–1633.

[27] J. Lin, C. Gan, and S. Han, TSM: Temporal shift module for efficient video understanding, in *Proc. IEEE/CVF Int. Conf. Computer Vision*, Seoul, Republic of Korea, 2019, pp. 7082–7092.

[28] J. Liu, X. Wang, C. Wang, Y. Gao, and M. Liu, Temporal decoupling graph convolutional network for skeleton-based gesture recognition, *IEEE Trans. Multimedia*, vol. 26, pp. 811–823, 2024.

[29] A. Shahroudy, J. Liu, T. T. Ng, and G. Wang, NTU RGB+D: A large scale dataset for 3D human activity analysis, in *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 1010–1019.

[30] J. Wang, X. Nie, Y. Xia, Y. Wu, and S. C. Zhu, Cross-view action modeling, learning, and recognition, in *Proc. 2014 IEEE Conf. Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014, pp. 2649–2656.

[31] B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, M. Aono, M. Burtscher, Q. Chen, N. K. Chowdhury, B. Fang, et al., A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries, *Comput. Vision Image Understanding*, vol. 131, pp. 1–27, 2015.

[32] Q. De Smedt, H. Wannous, and J. P. Vandeborre, Skeleton-based dynamic hand gesture recognition, in *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition Workshops*, Las Vegas, NV, USA, 2016, pp. 1206–1214.

[33] R. A. Torres-Guzman, M. R. Paulson, F. R. Avila, K. Maita, J. P. Garcia, A. J. Forte, and M. J. Maniaci, Smartphones and threshold-based monitoring methods effectively detect falls remotely: A systematic review, *Sensors*, vol. 23, no. 3, p. 1323, 2023.

[34] N. Gupta, S. K. Gupta, R. K. Pathak, V. Jain, P. Rashidi, and J. S. Suri, Human activity recognition in artificial intelligence framework: A narrative review, *Artif. Intell.*

Rev., vol. 55, no. 6, pp. 4755–4808, 2022.
[35] P. Kulurkar, C. K. Dixit, V. C. Bharathi, A. Monikavishnuvarthini, A. Dhakne, and P. Preethi, AI based elderly fall prediction system using wearable sensors: A smart home-care technology with IOT, *Meas.*: *Sens*., vol. 25, p. 100614, 2023.
[36] D. J. Warrington, E. J. Shortis, and P. J. Whittaker, Are wearable devices effective for preventing and detecting falls: An umbrella review (a review of systematic reviews), *BMC Public Health*, vol. 21, no. 1, p. 2091, 2021.
[37] M. Muaaz, S. Waqar, and M. Pätzold, Orientation-independent human activity recognition using complementary radio frequency sensing, *Sensors*, vol. 23, no. 13, p. 5810, 2023.
[38] Z. Chen, C. Cai, T. Zheng, J. Luo, J. Xiong, and X. Wang, RF-based human activity recognition using signal adapted convolutional neural network, *IEEE Trans. Mobile Comput*., vol. 22, no. 1, pp. 487–499, 2023.
[39] I. Jegham, A. Ben Khalifa, I. Alouani, and M. A. Mahjoub, Vision-based human action recognition: An overview and real world challenges, *Forensic Sci. Int.*: *Digital Invest.*, vol. 32, p. 200901, 2020.
[40] S. Yan, Y. Xiong, and D. Lin, Spatial temporal graph convolutional networks for skeleton-based action recognition, in *Proc. 32nd AAAI Conf. Artificial Intelligence*, New Orleans, Louisiana, USA, 2018, p. 912.
[41] L. Shi, Y. Zhang, J. Cheng, and H. Lu, Two-stream adaptive graph convolutional networks for skeleton-based action recognition, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 12018–12027.
[42] C. Bian, W. Feng, and S. Wang, Self-supervised representation learning for skeleton-based group activity recognition, in *Proc. 30th ACM Int. Conf. Multimedia*, Lisboa, Portugal, 2022, pp. 5990–5998.
[43] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, An attention enhanced graph convolutional LSTM network for skeleton-based action recognition, in *Proc. 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 1227–1236.
[44] L. Shi, Y. Zhang, J. Cheng, and H. Lu, Skeleton-based action recognition with directed graph neural networks, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 7904–7913.
[45] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, Semantics-guided neural networks for efficient skeleton-based human action recognition, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, pp. 1109–1118.
[46] X. Zhang, C. Xu, and D. Tao, Context aware graph convolution for skeleton-based action recognition, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, pp. 14321–14330.
[47] W. Peng, X. Hong, H. Chen, and G. Zhao, Learning graph convolutional network for skeleton-based human action recognition by neural searching, in *Proc. 34th AAAI Conf. Artificial Intelligence*, New York, NY, USA, 2020, pp. 2669–2676.
[48] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, Symbiotic graph neural networks for 3D skeleton-based human action recognition and motion prediction, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3316–3333, 2022.
[49] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, Actional-structural graph convolutional networks for skeleton-based action recognition, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 3590–3598.
[50] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, Channel-wise topology refinement graph convolution for skeleton-based action recognition, in *Proc. 2021 IEEE/CVF Int. Conf. Computer Vision*, Montreal, Canada, 2021, pp. 13339–13348.
[51] H. G. Chi, M. H. Ha, S. Chi, S. W. Lee, Q. Huang, and K. Ramani, InfoGCN: Representation learning for human skeleton-based action recognition, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, New Orleans, LA, USA, 2022, pp. 20154–20164.
[52] L. Shi, Y. Zhang, J. Cheng, and H. Lu, Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition, in *Proc. 15th Asian Conf. Computer Vision*, Kyoto, Japan, 2020, pp. 38–53.
[53] C. Plizzari, M. Cannici, and M. Matteucci, Spatial temporal transformer network for skeleton-based action recognition, in *Proc. Pattern Recognition. ICPR Int. Workshops and Challenges*, Virtual Event, 2021, pp. 694–701.
[54] H. Yang, D. Yan, L. Zhang, Y. Sun, D. Li, and S. J Maybank, Feedback graph convolutional network for skeleton-based action recognition, *IEEE Trans. Image Process.*, vol. 31, pp. 164–175, 2022.
[55] Y. F. Song, Z. Zhang, C. Shan, and L. Wang, Constructing stronger and faster baselines for skeleton-based action recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 1474–1488, 2023.
[56] Y. F. Song, Z. Zhang, and L. Wang, Richly activated graph convolutional network for action recognition with incomplete skeletons, in *Proc. 2019 IEEE Int. Conf. Image Processing*, Taipei, China, 2019, pp. 1–5.
[57] J. Guo, Q. Ji, and G. Shan, Overcomplete graph convolutional denoising autoencoder for noisy skeleton action recognition, *IET Image Processing*, vol. 18, no. 1, pp. 233–246, 2024.
[58] J. H. Song, K. Kong, and S. J. Kang, Dynamic hand gesture recognition using improved spatiotemporal graph convolutional network, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 6227–6239, 2022.
[59] X. S. Nguyen, L. Brun, O. Lézoray, and S. Bougleux, A neural network based on SPD manifold learning for skeleton-based hand gesture recognition, in *Proc. 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, CA, USA, 2019, pp. 12028–12037.

[60] J. Liu, Y. Liu, Y. Wang, V. Prinet, S. Xiang, and C. Pan, Decoupled representation learning for skeleton-based gesture recognition, in *Proc. 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, pp. 5750–5759.
[61] V. Veeriah, N. Zhuang, and G. J. Qi, Differential recurrent neural networks for action recognition, in *Proc. 2015 IEEE Int. Conf. Computer Vision*, Santiago, Chile, 2015, pp. 4041–4049.
[62] J. Wang, Z. Liu, Y. Wu, and J. Yuan, Learning actionlet ensemble for 3D human action recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 914–927, 2014.
[63] Y. Du, W. Wang, and L. Wang, Hierarchical recurrent neural network for skeleton based action recognition, in *Proc. 2015 IEEE Conf. Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015, pp. 1110–1118.
[64] M. Liu, H. Liu, and C. Chen, Enhanced skeleton visualization for view invariant human action recognition, *Pattern Recogn.*, vol. 68, pp. 346–362, 2017.
[65] I. Lee, D. Kim, S. Kang, and S. Lee, Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks, in *Proc. 2017 IEEE Int. Conf. Computer Vision*, Venice, Italy, pp. 1012–1020, 2017.
[66] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, and H. Lu, Decoupling GCN with DropGraph module for skeleton-based action recognition, in *Proc. 16th European Conf. Computer Vision*, Glasgow, UK, 2020, pp. 536–553.
[67] Y. F. Song, Z. Zhang, C. Shan, and L. Wang, Richly activated graph convolutional network for robust skeleton-based action recognition, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 5, pp. 1915–1925, 2021.
[68] C. Si, Y. Jing, W. Wang, L. Wang, and T. Tan, Skeleton-based action recognition with spatial reasoning and temporal stack learning, in *Proc. 15th European Conf. Computer Vision*, Munich, Germany, 2018, pp. 106–121.
[69] Z. Huang, X. Shen, X. Tian, H. Li, J. Huang, and X. Hua, Spatio-temporal inception graph convolutional networks for skeleton-based action recognition, in *Proc. 28th ACM Int. Conf. Multimedia*, Seattle, WA, USA, 2020, pp. 2122–2130.
[70] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition, in *Proc. 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, pp. 140–149.
[71] S. Wang, Y. Zhang, M. Zhao, H. Qi, K. Wang, F. Wei, and Y. Jiang, Skeleton-based action recognition via temporal-channel aggregation, arXiv preprint arXiv: 2205.15936, 2022.
[72] Z. Yang, K. Li, H. Gan, Z. Huang, and M. Shi, HD-GCN: A hybrid diffusion graph convolutional network, arXiv preprint arXiv: 2303.17966, 2023.
[73] H. Duan, J. Wang, K. Chen, and D. Lin, PYSKL: Towards good practices for skeleton action recognition, in *Proc. 30th ACM Int. Conf. Multimedia*, Lisboa, Portugal, 2022, pp. 7351–7354.
[74] L. Shi, Y. Zhang, J. Cheng, and H. Lu, Skeleton-based action recognition with multi-stream adaptive graph convolutional networks, *IEEE Trans. Image Process.*, vol. 29, pp. 9532–9545, 2020.
[75] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, BlazePose: On-device real-time body pose tracking, arXiv preprint arXiv: 2006.10204, 2020.

Vincent Gbouna Zakka is a PhD candidate at Aston University, UK. He received the BEng degree in electronics and information engineering from Liaoning University of Technology, China, and the MEng degree in mechatronics engineering from Zhejiang University, China. His research focuses on activity recognition, user trust in monitoring systems, vision-based monitoring technologies for home environments, and smart building technologies.

Luis J. Manso received the BEng degree in computer engineering in 2009 and the PhD degree in 2013, both from University of Extremadura, UK. He is currently a senior lecturer in computer science at Applied Artificial Intelligence and Robotics Department, Aston University, UK. His research interests include geometric learning, active perception, human-robot interaction, and sparse predictive world models.

Zhuangzhuang Dai received the BEng degree in electrical and electronics engineering from University of Birmingham, UK, the MEng and PhD degrees in propagation modeling for navigation and source location from University of Bath, UK. He is currently a lecturer in applied AI and robotics at Aston University, UK. Before joining Aston, he worked as a National Institute of Standards and Technology (NIST) software engineer at University of Oxford's Cyber-Physical Systems Group, UK, leading a project on "Pervasive, accurate, and reliable location based services for emergency responders". He also completed a knowledge transfer partnership project on smart RFID positioning systems at University of Manchester, UK. His research focuses on sensor fusion, embedded systems, deep learning, and human-robot interaction.