Spatio-temporal invariant descriptors for skeleton-based human action recognition

Kamel Aouaidjia [a], Chongsheng Zhang [a,*], Ioannis Pitas [b]

[a] School of Computer and Information Engineering, Henan University, Kaifeng, 475001, Henan, China
[b] Department of Informatics, Aristotle University of Thessaloniki, Greece

ARTICLE INFO

Keywords:
Skeleton-based action recognition
Spatio-temporal graph descriptor
Spatio-temporal motion descriptors
Invariant representations

ABSTRACT

Skeleton-based human action recognition is crucial for many practical applications. However, existing methods often rely on a single skeleton sequence representation, which may not fully capture the complex features of actions. To tackle this issue, we propose IMDAR (Invariant Multi-Descriptors for Action Recognition): A framework that uses multiple spatio-temporal invariant representations to improve action feature learning. These representations capture the evolution of skeleton poses, considering the motion of joints and limbs. We transform each skeleton in the sequence into a graph representation, and the sequence of graph features is structured into a spatio-temporal matrix. To capture the motion dynamics, we design three spatio-temporal distance matrices that represent the variation in inter-joint distances, inter-frame joint distances, and inter-limb angles across the sequence. The matrices are then transformed into image descriptors, which are used for training action prediction models. A Voting and Priority Score Selection (VPSS) algorithm is proposed to determine the correct class from multiple descriptor predictions. Experiments on benchmark datasets demonstrate the invariance capability of IMDAR, and show 2.4%, 1.3%, 1.8% and 2.8% improvement in accuracy on NTU-RGB+D 60, NTU-RGB+D 120, N-UCLA and UTD-MHAD datasets, respectively. Code and models are made available on the Github repository[1].

1. Introduction

Human action recognition is essential for various applications, such as human-computer interaction, video surveillance, robotics, healthcare, and virtual reality [11,42,5,29]. This topic has been investigated using different types of data, including RGB videos, depth videos, and body skeleton joints. Skeleton-based action recognition has gained significant attention because it is not affected by background, clothing, or lighting, and it provides precise body pose information in the form of 3D coordinates of the body joints. In the past, motion capture systems used wearable sensors, multiple cameras, or infrared sensors to capture accurate body pose data in a controlled environment.

Advancements in machine learning, deep learning, and artificial intelligence algorithms have profoundly transformed numerous fields, including healthcare, finance, forecasting, and computer vision [17–19,49,43]. In particular, deep learning models have advanced the ability to automatically extract hierarchical features from images and recognize complex patterns without manual feature

* Corresponding author.
E-mail addresses: kamel@henu.edu.cn (K. Aouaidjia), cszhang@ieee.org (C. Zhang), pitas@csd.auth.gr (I. Pitas).

[1] https://github.com/adjkamel/imdar

https://doi.org/10.1016/j.ins.2024.121832

engineering. This capability has led to significant progress in tasks like image classification, object detection, and segmentation, achieving human-level performance in many applications. As a result, these advancements have also made it easier to extract human body skeleton data from videos using deep learning methods, without relying on additional sensors [44].

Extracting distinctive features for each action from the skeleton sequence during the learning process is crucial for effective skeleton-based action recognition. Early hand-crafted methods extract features by converting the skeleton sequence into a simplified representation, which is then passed to a classifier, such as K-Nearest Neighbors, Random Forest, or a Hidden Markov Model [13,14]. With the advancements in deep learning for sequential data modeling, various network architectures have been proposed to predict actions from skeleton sequences, including Recurrent Neural Networks (RNNs) [12] and Long Short-Term Memory (LSTM) [25], which capture the relational features between skeleton joints over the sequence using attention mechanisms. Although, the skeleton data is often represented as 1D vector, a few methods have explored converting the input sequence into a 2D temporal representation, and then exploit the 2D feature extraction capability of Convolutional Neural Network (CNN) to learn action representations for class prediction [2].

Skeletons can be considered as graphs, where the vertices correspond to the skeleton joints and the edges represent the connections between joints. After the introduction of Graph Convolutional Networks (GCNs) [21], many recent methods use GCN baseline to capture spatio-temporal skeleton features in different ways [7,6]. The remarkable success of the Transformer [41] for modeling sequential data through the self-attention mechanism has inspired numerous studies to adopt the self-attention for skeleton action recognition. Transformer-based methods assign attention weights to different joints and frames, which allow the model to focus on the most representative spatial and temporal parts in each sequence [1,28]. In certain cases, attention-based methods are combined with Graph Convolutional Networks (GCNs) to achieve robust action representation [34,28].

A major drawback of existing methods is their dependence on a single representation that implicitly attempts to capture complex patterns. This approach can be challenging to control, as the feature extraction process becomes fixed once the model is trained, making it difficult to ensure that the model effectively captures all the action-specific features. This limitation becomes evident when actions that exhibit high similarity with other actions are misclassified. Although recent GCN and attention based methods have achieved state-of-the-art performance in skeleton-based human action recognition, they still have the following shortcomings: i) GCNs can only capture the feature dependencies between joints but overlook the motion of the joints. Moreover, as with over-smoothing issue [4], the relational representation between the joints (vertices) of different classes becomes more indistinguishable as the network deepens; ii) Although the attention mechanism can identify which parts of the sequence to focus on, it is sometimes biased and hard to control due to action diversity. iii) These methods seldom consider action invariance constraints, in particular, related to, e.g. the camera view angle and motion velocity.

To overcome the previous limitations, this paper proposes a new framework for skeleton-based action recognition, which is referred as IMDAR (**I**nvariant **M**ulti-**D**escriptors for **A**ction **R**ecognition). IMDAR consists of three modules; action representation, feature extraction, and action prediction. Specifically, in the action representation module, we devise five image-based spatio-temporal action descriptors for representing the action sequence from multiple perspectives to overcome the limited feature learning capability of single action representation. While there are many possible representations of human actions, our five proposed representations focus on key action dynamics to uniquely capture each action. When analyzing human body motion, actions are distinguished by the overall change in the skeleton pose over time and the direction of limb and joint movement. Movement direction can be represented in two complementary ways: through changes in the angles between limbs and changes in the distances between joints. Additionally, similar actions performed at different speeds will exhibit slight variations in angles and joint distances, which is why an invariant velocity representation is necessary.

To represent the skeleton pose evolution over time, we propose to transform the skeleton sequence into a spatio-temporal Graph Descriptor (GD), which is a feature map RGB image constructed from a graph matrix, where its columns represent the individual skeleton graph representations along the three Cartesian axes. Such representation is invariant to skeleton pose coordinates since it captures the change in the skeleton topological relationship between joints over time rather than relying specifically on the pose coordinates. The motivation behind this representation is that the over-smoothing problem in GCN-based methods makes the graph representation effectiveness depend on the network depth [4], while arranging the whole graph sequence once and for all in a fixed image provides a stable representation.

Since the graph representation fails to capture the dynamic motion of the joints, another three distance descriptors are proposed to track the spatio-temporal change in distance between joints and limbs across the sequence. Each descriptor is a single-channel grey image, obtained from a distance matrix, where its columns arrange the distance between joints or limbs. Specifically, the Joint Distance Descriptor (JDD) represents the inter-joint distances within each skeleton across the sequence. The Adjacent Distance Descriptor (ADD) represents the adjacent inter-frame joint distances along the sequence. It can visually represent the action velocity, allowing the same actions performed at different speeds to be classified as identical, which is beneficial for velocity invariance. The Limbs Angle Descriptor (LAD) captures the change in angles between adjacent limbs. It also provides insights about the direction of the joints based on the temporal change of angles. Finally, to create a global representation of the motion, the previous three distance descriptors are integrated into a three-channel Fusion Distance Descriptor (FDD). The four distance descriptors are view-invariant, as they only consider the distances between joints and limbs. Additionally, they represent actions of varying sequence lengths in a fixed-size image representation, which make them invariant to the number of frames in the sequence.

The feature extraction module consists of models trained to extract features from the five descriptors (GD, JDD, ADD, LAD and FDD). Since the descriptors are feature map images (Fig. 1) contain texture-like patterns, a shallow, low-level CNN model is designed to classify such patters. The features of each of the five descriptors are extracted independently through single-descriptor model for action prediction. Additionally, the features of the five descriptors are concatenated to form a robust and complementary action

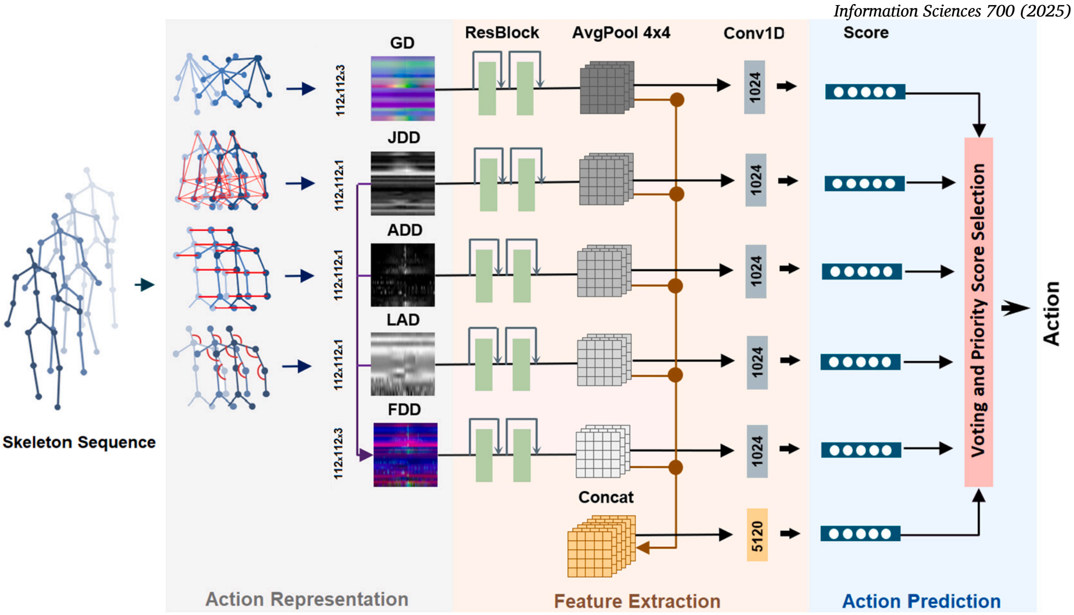

Fig. 1. The pipeline of IMDAR consists of three modules. The action representation module transforms the skeleton sequence into five image descriptors (GD: Graph Descriptor, JDD: Joint Distance Descriptor, ADD: Adjacent Distance Descriptor, LAD: Limbs Angle Descriptor, FDD: Fusion Distance Descriptor). The feature extraction module consists of a ResBlock and Average pooling (AvgPool 4 × 4). The action prediction module predicts the score from the six features (five descriptors and their concatenation) using 1D convolution (Conv1D), and then a VPSS algorithm is used to select the correct class.

representation using a fusion-model. In the prediction module, six predictions are obtained, i.e., five predictions from the single-descriptor models, and one prediction from the fusion-model. To select the correct predicted class, a simple and efficient Voting and Priority Score Selection (VPSS) algorithm is proposed to decide which prediction output should be considered as a final class. This algorithm is based on considering a majority voting in case of similar predictions and a pre-defined priority in the case of different predictions.

To evaluate the effectiveness of the proposed IMDAR framework, experiments have been conducted on four benchmark skeleton-based action recognition datasets. Namely, NTU-RGB+D 60, NTU-RGB+D 120, N-UCLA, and UTD-MHAD datasets. The results report that IMDAR outperforms most of state-of-the-art methods, and the ablation study validates its invariant representation capability. The main contributions of this paper can be summarized as follows:

(1) A proposed multiple spatio-temporal invariant image representations to address the limited feature learning capability of a single action representation.
(2) An introduced spatio-temporal graph descriptor to capture the change in the skeleton topological relationship between joints over time.
(3) To capture the motion dynamics, we design another four spatio-temporal distance descriptors that represent the variation in inter-joint distances, adjacent inter-frame joint distances, and inter-limb angles across the sequence.
(4) We present a voting and priority score selection algorithm that identifies the most probable correct action class among multiple predictions.

The rest of this paper is organized as follows: Section 2 reviews related works on skeleton-based action recognition. Section 3 introduces the technical details of the framework. Section 4 presents the analysis of the experimental results and ablation study, followed by a conclusion in Section 5.

2. Related work

2.1. CNN-based skeleton action recognition

There have been only few attempts to transform the skeleton sequence into 2D image representation and then use CNNs for feature extraction and classification. For example, Wang et al. [40] (Joint Trajectory Maps) transform the plot of the skeleton trajectories of the joints' motion from the top, side, and front views, into three images that are fed into three streams of CNNs for feature extraction and classification. Hernandez Ruiz et al. [15] encoded each skeleton into a 2D matrix which represents the Euclidean distance between each pair of joints, then the sequence of the 2D matrices is used as input to a 3D CNN for feature extraction and action prediction. In Caetano et al. [2] (Tree Structure Reference Joints Image), four joints are chosen as reference joints to create four corresponding images, each representing joint positions relative to one reference joint. These four images are stacked and processed by a CNN for action classification. De Boissiere and Noumeir [10] transformed the skeleton coordinates into an image that represents temporal changes in joint positions along the three Cartesian axes. Existing image representation methods ignore invariance to camera view angle and motion vilocity. Moreover, they only rely on a single representation that is unable to capture complementary features.

2.2. GCN-based skeleton action recognition

Following the success of Graph Convolutional Networks (GCNs) in modeling and classifying graph data, Yan et al. [46] introduced ST-GCN to model the spatial and temporal relationships among the joints in a skeleton sequence. This pioneering work marked the first successful application of GCNs for skeleton-based action recognition, and became the main component for numerous methods. Instead of using a manually defined skeleton topology, Shi et al. [32] (2s-AGCN) proposed an adaptive graph convolutional layer that learns the skeleton topology dynamically. Their approach employs a two-stream GCN for action prediction, one stream model the joints, while the second stream model the bones (second-order joints). Chen et al. [7] (MST-GCN) proposed multi-scale spatial (MS-GC) model that captures short-range joints dependencies and a multi-scale temporal (MT-GC) model which is capable of modeling long-range dependencies across the sequence. Kilis et al. [20] proposed a framework that first addresses missing joints in skeleton data through a feature imputation pre-processing step, and then employs a new adjacency matrix that treats the skeleton graph as clusters of nodes to enhance the performance. Instead of relying on a single skeleton topology, the CTR-GCN model of Chen et al. [6] learns multiple dynamic topologies to effectively aggregate joint features, then derives a shared topology from these learned representations for robust modeling. To tackle the issue of misclassifying ambiguous actions that are challenging to differentiate, Zhou et al. [50] (FRhead) proposed a Feature Refinement Head that discovers those actions and calibrates their features to be more distinguishable.

2.3. Attention-based skeleton action recognition

Several methods involve attention-based approaches, or combine attention with GCNs to enhance feature representation. To address the shortcomings of previous methods regarding the lack of spatial structural information and detailed temporal dynamics, Si et al. [35] (SR-TSL) introduced the SR-TSL model, which consists of a Spatial Reasoning Network (SRN) that captures spatial structure in each frame through a residual graph neural network and a Temporal Stack Learning Network (TSL) that models temporal dynamics using LSTM. Si et al. [34] (AGC-LSTM) integrate GCN and LSTM within a single network structure. This approach captures discriminative spatial and dynamic temporal features, and investigates co-occurrence relationship features between the spatial and temporal domains. Pang et al. [28] (IGFormer) addressed skeleton-based action interaction. The skeleton sequences of the two individuals are fed into a Semantic Partition Module (SPM) to model the interaction between body parts, then a Transformer is used for prediction. Bavil et al. [1] (Action Capsules) encoded global dependencies of joints effectively using the self-attention mechanism to concentrate on a specific set of joints for each action, then aggregate their features for action recognition. Since not all the skeleton joints are equally informative, Nikpour and Armanfard [27] proposed a method that removes the uninformative and misleading joints in each frame using on a trained deep reinforcement learning agent.

3. Methodology

The framework of IMDAR is shown in Fig. 1. It consists of three modules: action representation, feature extraction, and action prediction.

3.1. Action representation

We define the skeleton sequence as $Seq = \{S_k, k = 1, ..., T\}$, where T is the number of frames, and the skeleton $\mathbf{S}_k$ as a graph $\mathbf{S}_k = \{V, E\}$, $V = \{\{\{J_{i,j}\}_{i=1}^{N}\}_{j=1}^{3}$, $J_{i,j}$ is the coordinate j of the joint i, and N is the number of joints (vertices). Like De Boissiere and Noumeir [10], we normalize the skeleton sequence by considering the 'middle of the spine' joint of the first frame as the new origin of the coordinates system. In the rest of the coming sections, we use the term 'limb' to refer to a skeleton bone that forms an angle with its adjacent bone.

3.1.1. Skeleton graph matrix

For each skeleton $\mathbf{S}_k$ represented as $N \times 3$ matrix, the symmetric adjacency matrix is defined as $\mathbf{A} = \{a_{i,j}, i, j = 1, ..., N\}$, where $a_{i,j} = 1$ if the joints i and j are connected, and $a_{i,j} = 0$, otherwise. The normalized adjacency matrix $\mathbf{A}'$ is calculated by dividing each row by the sum of its values. The graph $\mathbf{G}_k$ of a single skeleton $\mathbf{S}_k$ is obtained by:

$$\mathbf{G}_k = \mathbf{S}_k^t \mathbf{A}' = \begin{bmatrix} j_{11} & j_{21} & \cdots & j_{N1} \\ j_{12} & j_{22} & \cdots & j_{N2} \\ j_{13} & j_{23} & \cdots & j_{N3} \end{bmatrix} \cdot \begin{bmatrix} a'_{11} & a'_{12} & \cdots & a'_{N1} \\ a'_{21} & a'_{22} & \cdots & a'_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ a'_{N1} & a'_{n2} & \cdots & a'_{NN} \end{bmatrix} = \begin{bmatrix} g_{11} & g_{21} & \cdots & g_{N1} \\ g_{12} & g_{22} & \cdots & g_{N2} \\ g_{13} & g_{23} & \cdots & g_{N3} \end{bmatrix}, \quad a'_{i,j} = a_{i,j} / \sum_{j=1}^{N} a_{i,j} \tag{1}$$

Where t represents the matrix transpose operation. The spatio-temporal graph representation of the sequence is given by the 3D graph matrix $\mathbf{M}^{gr} \in \mathbb{R}^{N \times 3 \times T}$ as:

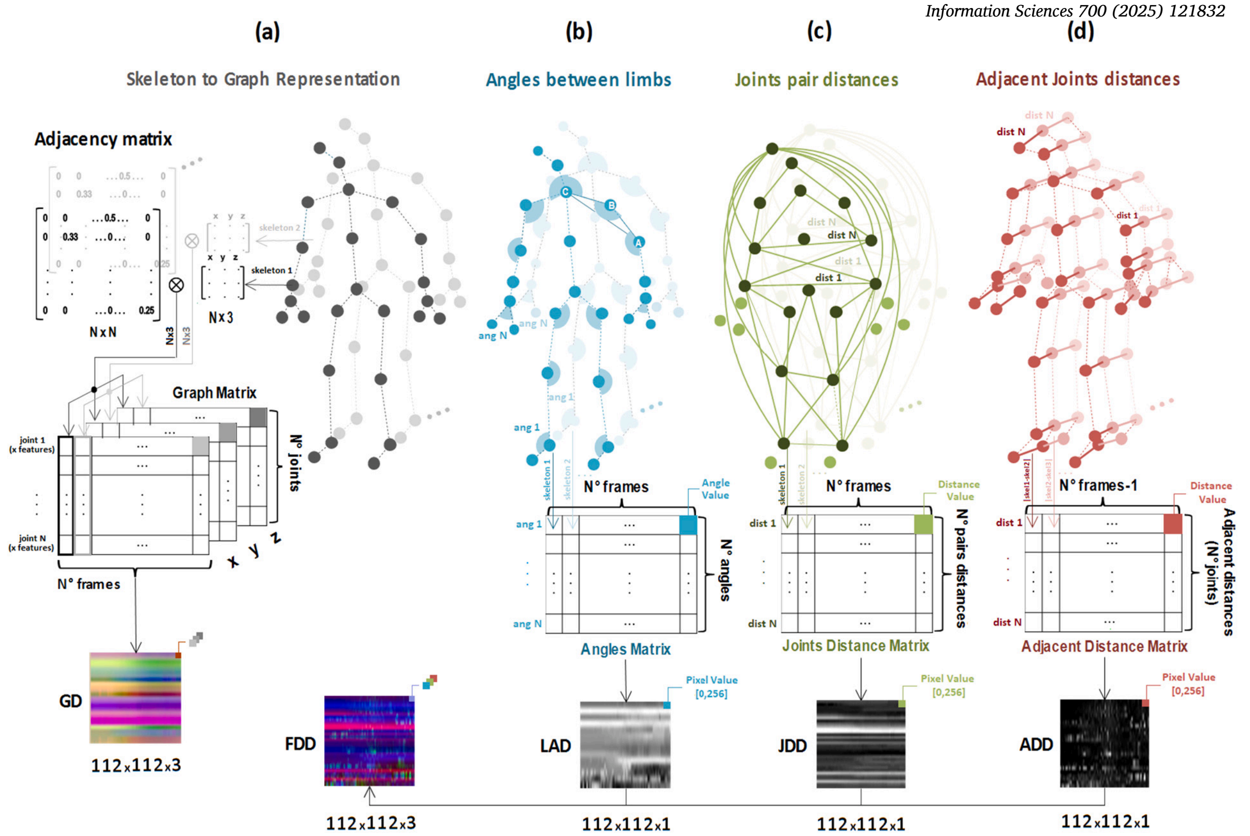

Fig. 2. The process of transforming the skeleton sequence into five spatio-temporal image descriptors. (a) GD: Graph Descriptor. (b) LAD: Limbs Angle Descriptor. (c) JDD: Joint Distance Descriptor. (d) ADD: Adjacent Distance Descriptor. FDD: Fusion Distance Descriptor.

$$\mathbf{M}^{gr} = \{\mathbf{G}_k^t\}_{k=1}^T = \{\{\{g_{i,j,k}\}_{i=1}^N\}_{j=1}^3\}_{k=1}^T = \begin{bmatrix} g_{1,j,1} & g_{1,j,2} & \cdots & g_{1,j,T} \\ g_{2,j,1} & g_{2,j,2} & \cdots & g_{2,j,T} \\ \vdots & \vdots & \ddots & \vdots \\ g_{N,j,1} & g_{N,j,2} & \cdots & g_{N,j,T} \end{bmatrix}, \quad j = 1,2,3 \tag{2}$$

Where $g_{i,j,k}$ are the features of the joint i of the coordinate j of the skeleton k. $\mathbf{M}^{gr}$ is a 3D matrix consists of three 2D matrices, corresponding to $j = 1,2,3$, which represents the three Cartesian axes. The construction of the graph matrix is illustrated visually in Fig. 2(a).

3.1.2. Joints distance matrix

One of the key factors for a robust motion representation is to capture the inter-joints distance evolution over time. For example, in the action ‘clapping’, it is important to know how far the left-hand joint is moving from the right-hand joint. To construct a joint distance representation, we selected the 32 most informative pairs that have a higher changeable rate during the motion than the other pairs Fig. 2(c). In the figure, the joints involved in the selected pairs are highlighted in dark green color.

Given a skeleton $\mathbf{S}_k$, we define the set of the selected pairs as: $Pair = \{p_i, i = 1,\ldots,P\}$, where P is the total number of selected pairs, and $p_i = (J^a, J^b)$ is the pair of the joints J^a and J^b. For each pair p_i, the Euclidean distance d_i is calculated between its two joints as: $d_i = \|J^a - J^b\|_2$. However, the distance between two joints in a skeleton of a shorter person is less than that of a taller person, due to the difference in limbs length. To normalize the distance d_i of the pair p_i for all body sizes, we divide the distance d_i by the body size z, where the normalized distance $d_i' = d_i/z$, and the body size $z = \sum_{i=1}^{N-1} L_i$, which is the sum of the lengths of all the skeleton limbs L_i, and the limb length is the distance between its joints. The joints distance matrix $\mathbf{M}^{jdis} \in \mathbb{R}^{P\times T}$ is defined as:

$$\mathbf{M}^{jdis} = \{\{d_i/z\}_{i=1}^P\}_{k=1}^T = \{\{d'_{i,k}\}_{i=1}^P\}_{k=1}^T = \begin{bmatrix} d'_{1,1} & d'_{1,2} & \cdots & d'_{1,T} \\ d'_{2,1} & d'_{2,2} & \cdots & d'_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ d'_{P,1} & d'_{P,2} & \cdots & d'_{P,T} \end{bmatrix} \tag{3}$$

Where d'_{ik} is the normalized distance of the pair p_i of the skeleton $\mathbf{S}_k$.

3.1.3. Adjacent distance matrix

The adjacent distance matrix represents the temporal change in the joint coordinates values between each two consecutive frames. In other words, the distance between the joint in a frame k and its new position in the frame $k+1$. It also represents action velocity, where larger distances indicate faster movement of the joints. Given two consecutive skeletons $\mathbf{S}_k$ and $\mathbf{S}_{k+1}$ of the sequence, the adjacent Euclidean distance $a_{i,k}$ of a joint i between two consecutive frames k and $k+1$ is written as: $a_{i,k} = \|J_{i,k} - J_{i,k+1}\|_2$. The Adjacent Distance Matrix $\mathbf{M}^{adis} \in \mathbb{R}^{N\times(T-1)}$ is defined as follows:

$$\mathbf{M}^{adis} = \{\{a_{i,k}\}_{i=1}^{N}\}_{k=1}^{T-1} = \{\{\|J_{i,k} - J_{i,k+1}\|_2\}_{i=1}^{N}\}_{k=1}^{T-1}$$
$$= \begin{bmatrix} \|J_{1,1} - J_{1,2}\|_2 & \|J_{1,2} - J_{1,3}\|_2 & \cdots & \|J_{1,T-1} - J_{1,T}\|_2 \\ \|J_{2,1} - J_{2,2}\|_2 & \|J_{2,2} - J_{2,3}\|_2 & \cdots & \|J_{2,T-1} - J_{2,T}\|_2 \\ \vdots & \vdots & \ddots & \vdots \\ \|J_{N,1} - J_{N,2}\|_2 & \|J_{N,2} - J_{N,3}\|_2 & \cdots & \|J_{N,T-1} - J_{N,T}\|_2 \end{bmatrix} \tag{4}$$

The calculation of $\mathbf{M}^{adis}$ is visually illustrated in Fig. 2(d).

3.1.4. Limbs angle matrix

The angle between two adjacent skeleton limbs provides information about how far the two limbs are moving from each other. However, such a feature cannot be captured by the previous distance representations, because even though the distance between the joints of the limbs is known, it is difficult to tell in which direction the joints are moving. The temporal change in the angles provides features that can represent the direction of the joints. As shown in Fig. 2(b), we selected 14 angles that have a higher changeable rate during the movement than the 4 remaining angles. However, all the angles could be considered, as an automatic selection.

Given two adjacent limbs AB and BC that share the same joint B (Fig. 2(b)), the angle $\theta = \widehat{ABC}$ of the triangle ABC can be calculated based on the coordinates values of the joints A, B, and C, using the law of Cosines:

$$AC^2 = AB^2 + BC^2 - 2\times AB\times BC\times\cos(\theta)$$
$$\theta = \arccos\left(\frac{AB^2 + BC^2 - AC^2}{2\times AB\times BC}\right) \tag{5}$$

Where AB, BC, and AC are the Euclidean distance between the joints A and B, B and C, and A and C, respectively:

$$AB = \|A - B\|_2, \quad BC = \|B - C\|_2, \quad AC = \|A - C\|_2 \tag{6}$$

Given a skeleton $\mathbf{S}_k$ of the sequence, we denote U as the number of the selected angles (Fig. 2(b)). We also denote $\theta_{i,k}$ as the angle i of the skeleton $\mathbf{S}_k$. The Limbs Angle Matrix $\mathbf{M}^{ang} \in \mathbb{R}^{U\times T}$ is defined as:

$$\mathbf{M}^{ang} = \{\{\theta_{i,k}\}_{i=1}^{U}\}_{k=1}^{T} = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,T} \\ \theta_{2,1} & \theta_{2,2} & \cdots & \theta_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{U,1} & \theta_{U,2} & \cdots & \theta_{U,T} \end{bmatrix} \tag{7}$$

3.1.5. Image descriptors

To construct a visual image representation of the four matrix representations, we transform the 3D graph matrix into a three-channel RGB image, and each of the three 2D distance matrices into a single-channel gray image. Before transforming the matrices into images, they are normalized to pixel values (between 0 and 255). We denote GD: Graph Descriptor, JDD: Joint Distance Descriptor, ADD: Adjacent Distance Descriptor, and LAD: Limbs Angle Descriptor, the transformation of the matrices $\mathbf{M}^{gr}$, $\mathbf{M}^{jdis}$, $\mathbf{M}^{adis}$, and $\mathbf{M}^{ang}$, respectively to images. Since the number of skeletons in the sequence varies according to the action, and to normalize the data for training and make it size-invariant, the images are resized to 112 × 112. Choosing such a larger size than the matrices size is convenient to preserve features. To obtain a robust representation that captures the spatio-temporal motion of the joints distance and limbs together, we construct an RGB descriptor FDD (Fusion Distance Descriptor) by stacking the three gray descriptors (JDD, ADD, and LAD). The transformation of the matrices into image descriptors can be formalized as follows:

$$\begin{aligned} Norm(\mathbf{M}) &= 255\times\frac{\mathbf{M} - \min(\mathbf{M})}{\max(\mathbf{M}) - \min(\mathbf{M})}, \\ Desc(M) &= Resize(Img(Norm(M)), 112\times 112), \\ \{GD, JDD, ADD, LAD\} &= \{Desc(\mathbf{M}^{gr}), Desc(\mathbf{M}^{jdis}), Desc(\mathbf{M}^{adis}), Desc(\mathbf{M}^{ang})\}, \\ FDD &= Stack(JDD, ADD, LAD). \end{aligned} \tag{8}$$

Where Img is the function that transforms a matrix to image. Some of the actions require two people in the sequence like ‘shaking hands’. Inspired by [10], our proposed method can handle two people in the sequence by doubling the number of joints N, the number of selected pairs P, the number of joints N, and the number of angles U in $\mathbf{M}^{gr}$, $\mathbf{M}^{jdis}$, $\mathbf{M}^{adis}$, and $\mathbf{M}^{ang}$, respectively. Fig. 2 illustrates the construction of the matrix representations and the image descriptors. Fig. 4 shows the result of constructing the five descriptors in case of one or two people performing the action.

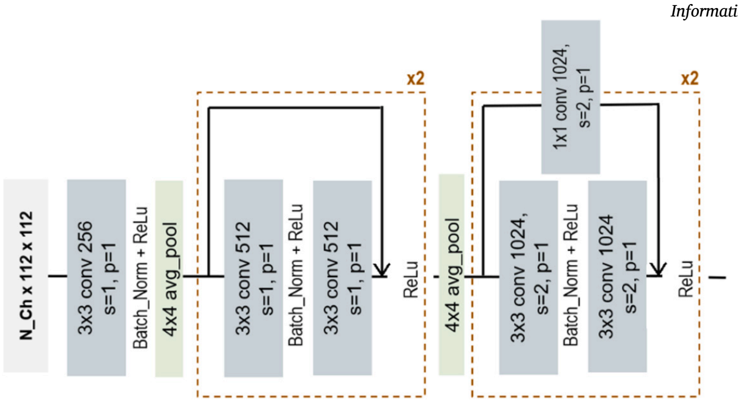

Fig. 3. A detailed structure of the ResBlock in Fig. 1. N_ch: number of channels, s: stride, p: padding.

Algorithm 1 Voting and Priority Score Selection.

```
Input: $c^{GD}$, $c^{JDD}$, $c^{ADD}$, $c^{LAD}$, $c^{FDD}$, $c^{cat}$
Output: class
Priority: $\Pr(c^{cat}) > \Pr(c^{GD}) > \Pr(c^{FDD}) > \Pr(c^{JDD}) > \Pr(c^{ADD}) > \Pr(c^{LAD})$
Same Predictions: Sm = Same($c^{GD}$, $c^{JDD}$, $c^{ADD}$, $c^{LAD}$, $c^{FDD}$, $c^{cat}$)
if Sm > = 2 then // two or more predictions have the same class
    class = any(Sm)
else if Sm = (3,3) or Sm = (2,2,2) then // different classes predicted equally
    class = Max(Pr(Sm(.)))
else
    $P$: set of all pairs of the predicted classes $(c^a, c^b)$, where $Pr(c^a) > Pr(c^b)$
    if $\Pr(c^a)$ > all($\Pr(c^a)$) and $\Pr(c^b)$ > all($\Pr(c^b)$) then
        class = $c^b$
    end if
end if
```

3.2. Feature extraction and action classification

Fig. 1 shows the feature extraction module of IMDAR. Each of the five descriptors is used as input to a ResBlock network (Fig. 3), then a 4×4 average pooling is applied on the output of the ResBlock to obtain features of size 1×1×1024. After that, a 1D Convolution is applied to get the prediction score vector with a size equal to the number of classes. For complementary feature representation of the action, the outputs of the average pooling layers of the five descriptors are concatenated to obtain features of size 1 × 1 × 5120, then a 1D Convolution is applied to get a prediction score of the concatenated features. The ResBlock (Fig. 3) is carefully designed to fit our image patters by using few layers for low-level feature extraction. The feature extraction module can be formulated as follows:

$$\begin{aligned} D &= \{GD, JDD, ADD, LAD, FDD\} \\ f^D &= AvgPool(ResBlock(D), 4 \times 4) \\ c^D &= Conv(f^D, 1024, 1 \times 1) \\ f^{cat} &= Cat(f^{GD}, f^{JDD}, f^{ADD}, f^{LAD}, f^{FDD}) \\ c^{cat} &= Conv(f^{cat}, 5120, 1 \times 1) \end{aligned} \tag{9}$$

Where f^D are the features of the descriptor D, and f^{cat} are the concatenated features of the five descriptors. c^D is the predicted class from the descriptor D, and c^{cat} is the predicted class from the concatenation of the five descriptors. $Conv$, $AvgPool$, and Cat are the 1D convolution, the average pooling, and the concatenation operations, respectively.

We denote $Pred = \{c^{GD}, c^{JDD}, c^{ADD}, c^{LAD}, c^{FDD}, c^{cat}\}$, the predicted classes from the six models. The models generate different predictions for the same input sequence, and some predictions like c^{GD} and c^{JDD} can be more accurate. To decide which of the six class predictions must be considered as the final class, we propose a Voting and Priority Score Selection (VPSS) algorithm (Algorithm 1). The idea of the algorithm is to assign a priority to each of the models based on its individual performance:

$$Pr(c^{cat}) > Pr(c^{GD}) > r(c^{FDD}) > Pr(c^{JDD}) > Pr(c^{ADD}) > Pr(c^{LAD}) \tag{10}$$

Where Pr indicates the priority. If two or more models predictions agree on a class, we consider it as the final predicted class:

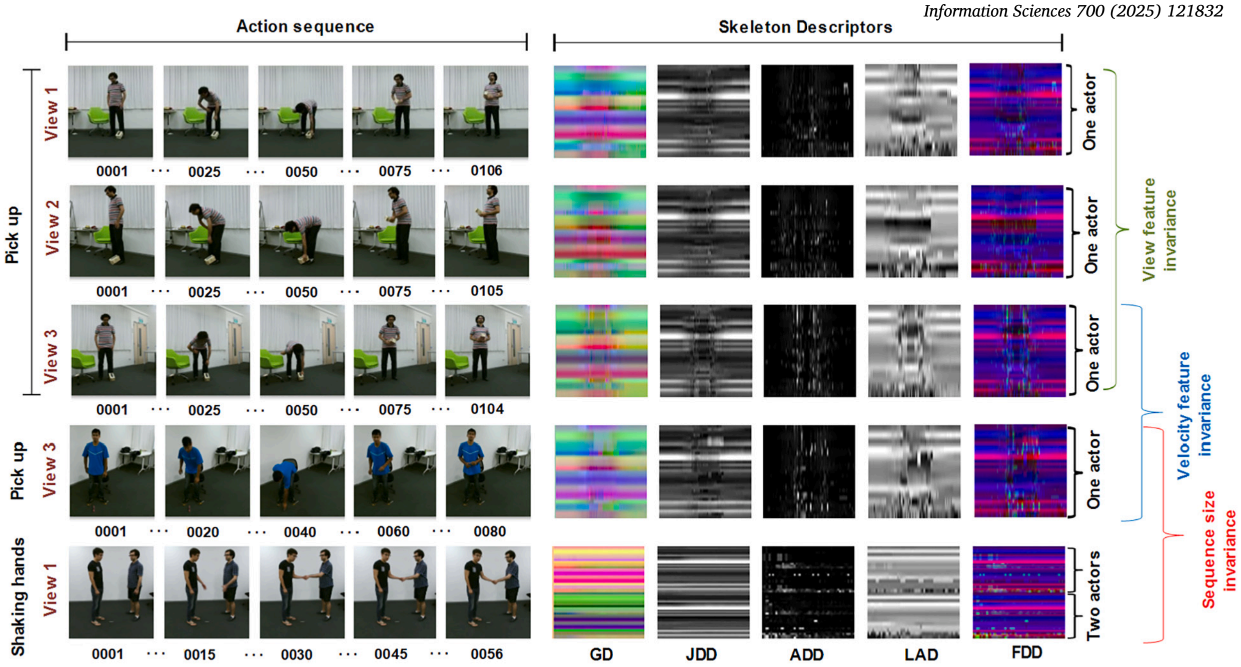

Fig. 4. Visualization of the action representation and the feature invariance with different views, velocities, sequence sizes, and number of actors. Left: key video frames of the actions. Right: Action representation via image descriptors.

$$
\begin{aligned}
&P_{\text{vote}} = Vote(Pred) \\
&\text{class} = majority(Pred) \quad \text{if} \quad |P_{\text{vote}}| \geq 2
\end{aligned}
\tag{11}
$$

If some of the models agree on one class and others agree on another class equally, we trust the models that the sum of their priority is higher:

$$
\text{class} = \arg \max_{c^k \in Pred} \sum_{i \in agree(c^k)} Pr(c^i)
\tag{12}
$$

Where $agree(c^k)$ represents the set of models that predict the class c^k. In the rest of the cases when the models don't agree on the same class, we consider the ones with less priority but not the least priority, because sometimes the correct class can be predicted with the less performant model. For the latter case, we create a combination of pairs (c^a, c^b) from the predicted classes, where the first element of the pair has higher priority than the second element, then we select the pair that both its elements are higher than all the elements of the rest of the pairs, and the correct class corresponds to the second element of the pair:

$$
\text{class} = \arg \max_{(c^a, c^b)} \left(Pr(c^a) > Pr(c^b) \right) = c^b
\tag{13}
$$

4. Experiments

4.1. Datasets

The NTU-RGB+D 60 is a large-scale human action recognition dataset with 60 action classes, containing 56,880 skeleton sequences. For comparison with the state-of-the-art, we follow the same evaluation protocol of the dataset [31]: Cross-Subject (C-Sub) and Cross-View (C-View). The NTU-RGB+D 120 dataset is an extension of NTU-RGB+D 60 with 60 additional actions. It contains 114,480 sequences across 120 classes. We also follow the same evaluation protocol of the dataset [24]: Cross-Subject (C-Sub) and Cross-Setup (C-Set). The N-UCLA (Northwestern-UCLA) dataset consists of 1,494 sequence clips of 10 actions captured from 3 camera angles simultaneously. In the testing benchmark [39], camera views 1 and 2 are used for training, and camera view 3 is used for testing. UTD-MHAD [3] dataset includes 861 sequences of 27 actions. The evaluation protocol for this dataset is Cross-Subject [22].

4.2. Implementation details

Six models are trained for action prediction (Fig. 1). Five models are trained separately for each descriptor, and the sixth model (fusion-model) is trained using the five descriptors together. All the six models are trained with 0.0001 learning rate that decreases by 0.5 after the fifth epoch, using Adam optimizer with a batch size of 32. A Cross-Entropy loss is used to train each of the five models, but for the fusion-model, the total loss is the sum of eight Cross-Entropy losses: five losses from the individual models, one

Table 1
Comparison of accuracy between IMDAR and the state-of-the-art action recognition methods on the NTU-RGB+D 60 DATASET for Cross-Subject (C-Sub) and Cross-View (C-View) benchmarks (%).

Method	C-Sub	C-View
ST-GCN [46] (AAAI'18), GCN	81.5	88.3
SR-TSL [35] (ECCV'18), GNN+Att	84.8	92.4
AS-GCN [23] (CVPR'19), GCN	86.8	94.2
2s-AGCN [32] (CVPR'19), GCN	88.5	95.1
AGC-LSTM [34] (CVPR'19), GCN+Att	89.2	95.0
RA-GCN [36] (TCSVT'20), GCN	87.3	93.6
4s-Shift-GCN [8] (CVPR'20), GCN	90.7	96.5
Dynamic-GCN [47] (ACMMM'20), GCN	91.5	96.0
MST-GCN [7] (AAAI'21), GCN	91.5	96.6
ST-TR [30] (CVIU'21), GCN+Att	90.3	96.3
Ta-CNN [45] (AAAI'22), CNN	90.7	95.1
EfficientGCN [37] (TPAMI'22), GCN+Att	92.1	96.1
ST-SLKA [26] (TVCG'23), GCN+Att	90.7	96.1
Action Capsules [1] (CVIU'23), Att	90.0	96.3
SHARL [27] (TSMCS'23), Att	90.4	96.5
IMDAR (Proposed)	**92.8**	**96.8**

Table 2
Comparison of accuracy between IMDAR and the state-of-the-art action recognition methods on NTU-RGB+D 120 DATASET for Cross-Subject (C-Sub) and Cross-Setup (C-Set) benchmarks (%).

Method	C-Sub	C-Set
ST-GCN [46] (AAAI'18), GCN	70.7	73.2
AS-GCN [23] (CVPR'19), GCN	77.9	78.5
2s-AGCN [32] (CVPR'19), GCN	82.5	84.2
RA-GCN [36] (TCSVT'20), GCN	81.1	82.7
4s-Shift-GCN [8] (CVPR'20), GCN	85.9	87.6
Dynamic-GCN [47] (ACMMM'20), GCN	87.3	88.6
MST-GCN [7] (AAAI'21), GCN	87.5	88.8
ST-TR [30] (CVIU'21), GCN+Att	85.1	87.1
Ta-CNN [45] (AAAI'022), CNN	85.7	87.3
IGFormer [28] (ECCV'22), Att	85.4	86.5
EfficientGCN [37] (TPAMI'22), GCN+Att	**88.7**	88.9
ST-SLKA [26] (TVCG'23), GCN+Att	86.3	87.8
IMDAR (Proposed)	**87.5**	**89.1**

loss from the concatenated features of all the models, one loss from the concatenated features of GD and FDD, and one loss from the concatenated features of JDD, ADD and LAD. The experiments were conducted using PyTorch framework on a machine with Nvidia RTX 4090 GPU, 64 GB of RAM, and a CPU of Intel(R) Core(TM) i9-13900k.

4.3. Comparison with the state-of-the-art

The comparison with the state-of-the-art methods on NTU-RGB+D 60 for both C-Sub and C-View benchmarks are shown in Table 1. IMDAR demonstrates higher prediction accuracy on the C-Sub benchmark and performs even better than methods that combine both GCN and attention mechanisms. Specifically, it shows better performance than SHARL [27] by 2.4% (C-Sub), which focuses on dynamic joint selection through reinforcement learning, while our multiple invariant representations provide a comprehensive encoding of actions that can adapt to any dynamic change without the need for complex training frameworks.

The performance on the NTU-RGB+D 120 dataset in Table 2 indicates that our work is on par with state-of-the-art methods on the C-Sub benchmark and surpasses the new state-of-the-art methods on the C-Set benchmark. IMDAR outperformed IGFormer [28] by 2.1% on C-Sub and 2.6% on C-Set. IGFormer focuses on human interaction recognition using graph representations for each person, combined with self-attention. Unlike this method, our representations can effectively capture interactions among people through a 2D representation for both skeletons. Additionally, the integration of multiple representations, including a spatio-temporal graph representation, enhances the model's ability to process complex interactions.

Further comparison on a smaller dataset such as N-UCLA is shown in Table 3. Although FRHead [50] includes an auxiliary feature refinement module, which focuses on dynamic calibration of ambiguous samples through a specialized refinement head, our method shows better performance by 2.3%. This improvement is attributed to the multiple representations that provide diverse features to distinguish actions that are highly similar to other actions, helping to reduce ambiguity. InfoGCN [9] combines an information bottleneck-based learning objective with attention-based graph convolution to learn effective latent representations. However, IMDAR

Table 3
Accuracy comparison of IMDAR with the state-of-the-art action recognition methods on N-UCLA DATASET for Cross-View benchmark (%).

Method	Accuracy
AGC-LSTM [34] (CVPR'19), Att	93.3
VA-CNN [48] (TPAMI'19), CNN+Att	90.7
4s-shift-GCN [8] (CVPR'20), GCN	94.6
Ta-CNN [45] (AAAI'22), CNN	96.1
CTR-GCN [6] (CVPR'22), GCN	96.5
InfoGCN [9] (CVPR'22), GCN+Att	96.6
FRHead [50] (CVPR'23), GCN	96.8
Action Capsules [1] (CVIU'23), Att	97.3
IMDAR (Proposed)	**99.1**

Table 4
Accuracy comparison of IMDAR with action recognition methods on UTD-MHAD DATASET (%) on Cross-Subject benchmark (%).

Method	Accuracy
Kinect [3] (ICIP'15)	66.1
Inertial [3] (ICIP'15)	67.2
Kinect&Inertial [3] (ICIP'15)	79.1
JTM [40] (ACMMM'16)	85.8
Optical Spectra [16] (TCSVT'16)	86.9
JDM [22] (SPL'17)	88.1
IMDAR (Proposed)	**97.3**

Table 5
Statistical analysis of P-Values across datasets. P-Values below 5×10^{-2} threshold indicate significant improvement in accuracy. For each dataset, the accuracies of the compared models are used as baselines.

Dataset	Nbr of baselines	P-Values	Accuracy (%)
NTU-RGB+D 60 (C-Sub)	16	7.0×10^{-5}	92.8
NTU-RGB+D 60 (C-View)	16	2.4×10^{-3}	96.8
NTU-RGB+D 120 (C-Sub)	13	1.1×10^{-2}	87.5
NTU-RGB+D 120 (C-Set)	13	6.6×10^{-3}	89.1
N-UCLA	8	1.8×10^{-3}	99.1
UTD-MHAD	7	2.7×10^{-3}	97.3

shows better accuracy by 2.5%, which can be attributed to the fact that the action descriptors are more effective than the latent representation.

In Table 4, we present the prediction accuracy on UTD-MHAD dataset. IMDAR achieves 9.2% improvement over the latest skeleton-based action recognition method tested on this dataset (JDM [22]), which uses multiple coordinate-based representations of the skeleton sequences. This significant improvement demonstrates the effectiveness of our diverse spatio-temporal graph-based and distance-based representations.

Table 5 presents the p-values for each dataset to statistically analyze the improvement in accuracy of IMDAR over existing methods mentioned in Tables 1, 2, 3, and 4. Overall, the p-values are below 5×10^{-2} threshold, which indicates significant improvement. Especially for NTU-RGB+D 60 (C-Sub) and N-UCLA datasets.

In Fig. 5(left), the confusion matrix for the Cross-Subject benchmark on NTU-RGB+D 60 shows the prediction accuracy for each class, with 20 actions displayed numerically and the rest represented by cell colors. Misclassified actions are highlighted with red rectangles, and while overall accuracy is high, slight confusion occurs for actions with similar movements, such as '11-reading' and '12-writing' being confused with '29-play with phone/tablet' and '37-salut'. Fig. 5(right) for the Cross-View benchmark presents minimal confusion, confirming the effectiveness of view invariance. As the action classes increase to 120 in NTU-RGB+D 120 dataset, the accuracy remains stable for both Cross-Subject and Cross-Setup benchmarks (Fig. 6). For Cross-Subject, '76-cutting paper' is confused with actions like '73-stable book' and '74-counting money', while '83-ball up paper' overlaps with several other actions. In Cross-Setup, actions like '71-make OK sign' and '75-cutting nails' are confused with '104-stretch oneself' and '86-apply cream on hand'.

4.4. Ablation studies

4.4.1. Effect of each component on the prediction

Each descriptor represents one aspect of the action. For some actions, one descriptor is enough to predict the correct class. However, for complex actions, multiple descriptors are needed to have a complete representation using different features. The performance of

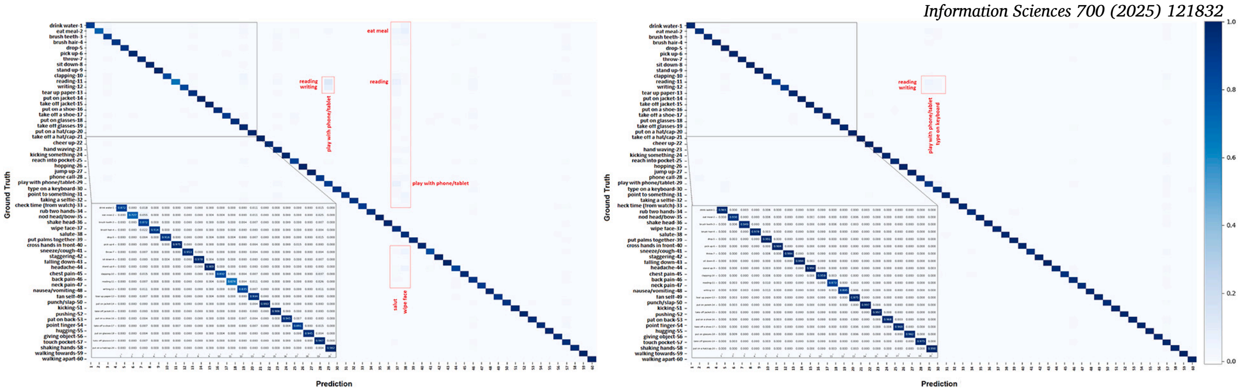

Fig. 5. Confusion matrix of the class prediction on NTU-RGB+D 60 dataset for Cross-Subject benchmark (left) and Cross-View benchmark (right). The areas highlighted in red indicate the predicted actions that have been misclassified as other actions.

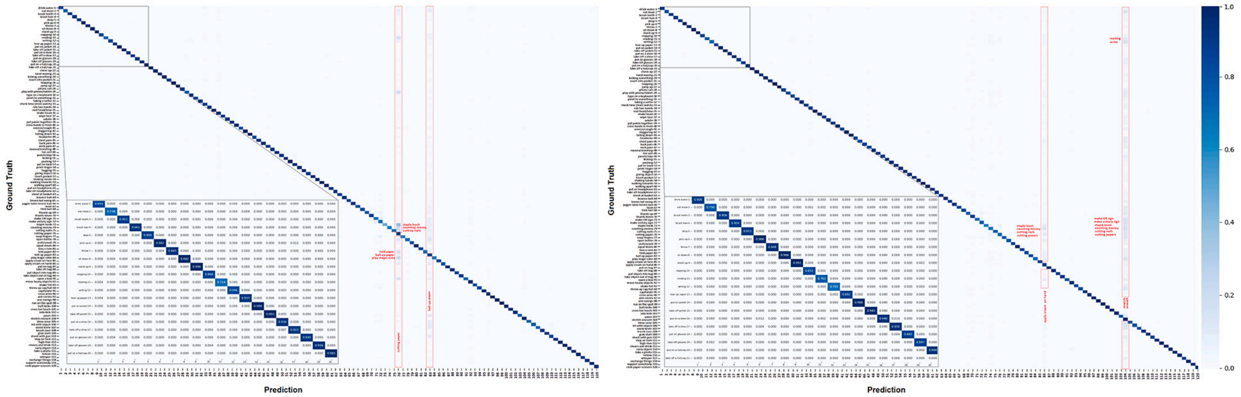

Fig. 6. Confusion matrix of the class prediction on NTU-RGB+D 120 dataset for Cross-Subject benchmark (left) and Cross-Setup benchmark (right). The areas highlighted in red indicate the predicted actions that have been misclassified as other actions.

Table 6
Accuracy of single-descriptor models, the fusion-model, and whole-framework on different datasets (%) .

Models	NTU-RGB+D 60		NTU-RGB+D 120		N-UCLA
	C-Sub	C-View	C-Sub	C-Set	
GD	80.3	86.2	72.0	74.8	83.8
JDD	77.8	86.4	68.0	70.8	91.0
ADD	73.6	80.1	62.2	65.2	88.1
LAD	73.1	77.9	61.4	63.7	87.7
FDD	81.1	87.7	72.1	74.4	92.1
fusion-model	85.2	91.2	76.7	79.2	95.0
whole-framework	92.8	96.8	87.5	89.1	99.1

single-descriptor models on each dataset for different benchmarks is shown in Table 6. The priority given to each model when applying the VPSS algorithm is based on the results of the table. The VPSS has a big effect on the final prediction accuracy. It is better than the fusion-model on the NTU-RGB+D 60 dataset by 7.6% and 5.7% for the C-Sub and C-View benchmarks, respectively. The impact is even higher than the fusion-model on the NTU-RGB+D 120 dataset by 10.8% and 9.9% for the C-Sub and C-Set, respectively. Applying VPSS increases the prediction performance even more to 99.1% on N-UCLA dataset.

To evaluate the impact of each descriptor on the overall framework, Table 7 shows the accuracy achieved when one or two descriptors are omitted. The results indicate that omitting a single descriptor leads to a drop in accuracy, while omitting two descriptors results in an even greater decline. This demonstrates how each descriptor contributes to the effectiveness of the proposed representations when used together. Specifically, the absence of the GD or the fusion-model results in a decrease of 2.2% and 2.4%, respectively, which both have a higher impact compared to the absence of JDD, ADD, LAD or FDD with a decrease of just 1.1%, 1.6%, 1.1%, and 1%, respectively. However, the absence of GD and JDD, ADD and LAD, or FDD and fusion-model lead to a higher decrease of 4.7%, 3.4%, and 4.2%, respectively.

Table 7
Ablation study on the effect of each descriptor on the global accuracy. The results are obtained on NTU-RGB+D 60 for C-Sub benchmark (%).

GD	JDD	ADD	LAD	FDD	fusion-model	Accuracy
		✓	✓	✓	✓	88.1
✓	✓			✓	✓	89.4
✓	✓	✓	✓			88.6
	✓	✓	✓	✓	✓	90.6
✓		✓	✓	✓	✓	91.7
✓	✓		✓	✓	✓	91.2
✓	✓	✓		✓	✓	91.7
✓	✓	✓	✓		✓	91.8
✓	✓	✓	✓	✓		90.4
✓	✓	✓	✓	✓	✓	92.8

Table 8
Comparison of accuracy between single-descriptor model and Multi-Axis Vision Transformer (MaxVIT) model on NTU-RGB+D 60 (C-Sub) and N-UCLA datasets (%).

	NTU-RGB+D 60		N-UCLA	
Descriptors	Ours	MaxVIT	Ours	MaxVIT
GD	**80.3**	74.4	**83.8**	73.1
JDD	**77.8**	60.1	**91.0**	87.3
ADD	**73.6**	68.2	**88.1**	78.3
LAD	**73.1**	64.6	**87.7**	82.1
FDD	**81.1**	71.5	**92.1**	88.5

4.4.2. Classification with a vision Transformer

Vision Transformers exceeded state-of-the-art CNN models for image classification. Table 8 shows the classification results on NTU-RGB+D 60 dataset for the C-Sub benchmark and N-UCLA dataset, compared to the prediction results using a state-of-the-art vision Transformer MaxVIT [38] to classify our descriptors. Our simple CNN model exceeded the Transformer by a large difference in accuracy. The comparison results confirm our first assumption which stated that vision Transformers may not be efficient to classify such kind of feature map images, and verify the efficacy of the ResBlock with its shallow structure. A shallow CNN model is more suitable to classify such images, because it primarily captures low-level features, such as edges, textures, and simple patterns. In contrast, vision Transformers are designed to capture high-level semantic features and global context.

4.4.3. View invariance

The descriptors JDD, ADD, and LAD are distance-based, which means that their representations are independent of the skeleton pose coordinates. For example, the same action viewed from different angles has different skeleton coordinates, but the distances between joints remain unchanged. These four distance descriptors significantly impact view invariance and action prediction. Moreover, since the GD descriptor represents the changes in the skeleton joints topological relationships, it makes it view-independent. In Fig. 5, we show view-invariant features of the 'pick up' action of the NTU-RGB+D 60 dataset from three different angles. The visual comparison of five feature maps confirms the invariance of the action across views, though small differences in ADD descriptors due to speed variations between adjacent frames.

For quantitative evaluation of the view invariance, in Fig. 7, we show the prediction accuracy of individual classes on NTU-RGB+D 60 for C-Sub benchmark on three different views for each action. The figure reports that the accuracy for each action from the three views is stable, i.e., the performance depends on the action type, and not on the view. In most cases, for the same action, if the accuracy is high on a certain view, it is also high with almost the same value on the other views, and vice versa. The view invariance is reflected in having the same graph shape on the three views.

4.4.4. Velocity and sequence size invariance

To evaluate the performance of the descriptors on different action velocities, in Fig. 8, we visualize in detail the accuracy of each action, categorized based on the number of frames in the samples. The number of frames in the video samples are organized in three intervals: [36-100], [100-150], and [150-200]. For the same action, samples with large velocity contain few frames, while samples with small velocity contain larger number of frames. We notice that the accuracy is not affected by the difference in the number of frames. For example, for the action 'wipe face' the accuracy is 93.12% despite having actions of three different ranges, while the accuracy for the action 'clapping' is just 83.46% despite having one range of frames. Even though the action 'put on a shoe' has the largest number of actions with two very different ranges, its accuracy (94.51%) still better than 'drop' (91.64%) which contain only one range of frames. The reason for this performance can be attributed to the fact that the representations of actions using a fixed-size images of 112 × 112 ensure sequence size invariance. Additionally, the ADD captures the movement of joints between adjacent frames, which contributes to the overall representation by incorporating velocity features.

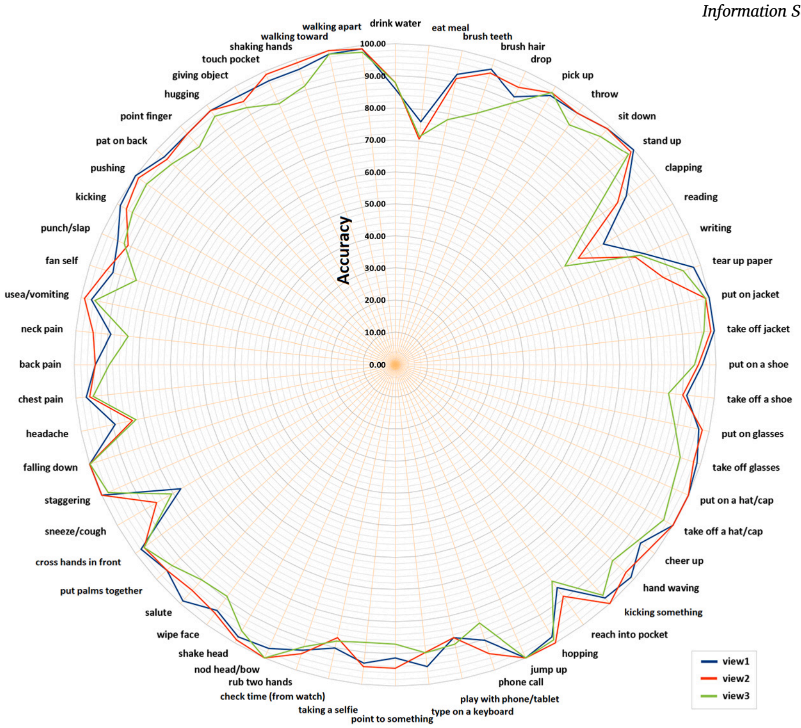

Fig. 7. Analysis of the view invariance of IMDAR for each class across the three views in the NTU-RGB+D 60 dataset for the C-Sub benchmark. Overall, the prediction accuracy for each action across the three views are closely aligned with each other.

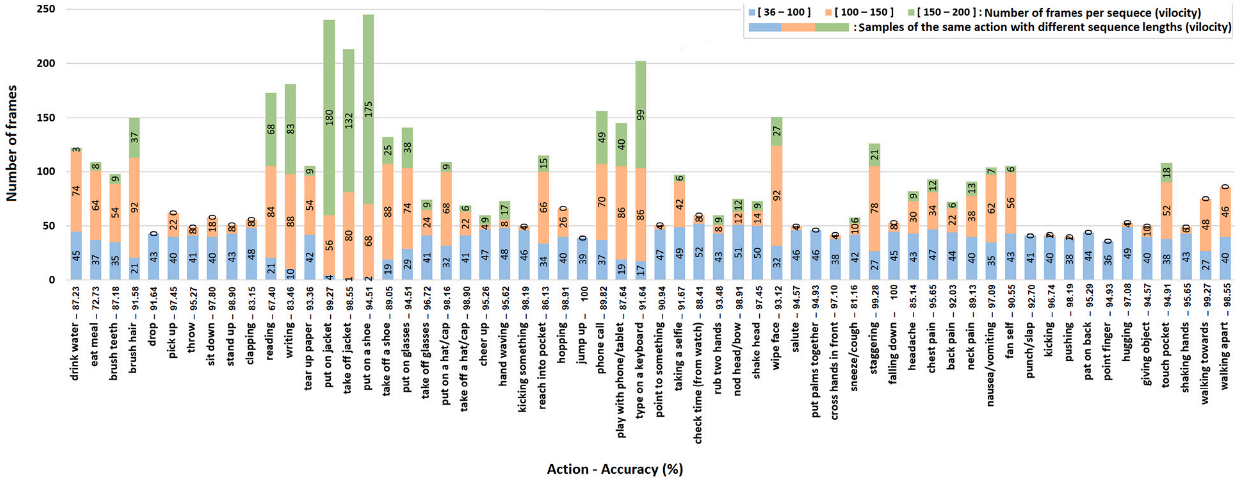

Fig. 8. Analysis of the velocity invariance of IMDAR. Each action contain video samples with different lengths. The length of video samples of each action are organized into three categories: [36,100], [100,150], and [150,200]. The results are obtained on NTU-RGB+D 60 for C-Sub benchmark. It is clearly observed that IMDAR generates high prediction accuracy for actions of varying video lengths.

4.5. Computation complexity

Table 9 compares IMDAR with other action recognition models on FLOPS (Floating Point Operations Per Second) and inference time (sequences/second). The single-descriptor model shows 7.8 FLOPS, ranking second. In contrast, the fusion-model requires 31.6 FLOPS, which is approximately five times the FLOPS of the single-descriptor model due to the concatenated features of the five descriptors. Since the entire framework consists of multiple models, we calculate its FLOPS by summing the FLOPS of all individual models, resulting in a total of 70.6 FLOPS. From Table 1 and Table 3, we observe that IMDAR (whole-framework) significantly outperforms the methods listed in Table 9, despite a slightly higher computational demand. Furthermore, the performance of the fusion-model, which requires 36.1 FLOPS, is comparable to other methods such as ST-GCN [46] and SR-TSL [35] in Table 1. Regarding

Table 9
Computation complexity comparison of IMDAR models with other action recognition models on Floating-Point Operations Per Second (FLOPS) and inference time (sequence/second).

Models	FLOPS	Inference time
ST-GCN [46]	16.3	42.91
RA-GCN [36]	32.8	18.72
2s-AGCN [32]	37.3	22.31
4s-ShiftGCN [8]	10.0	-
CTR-GCN [6]	7.6	-
DSTA-Net [33]	64.7	-
ST-TR [30]	259.4	-
IMDAR (single-descriptor)	7.8	108
IMDAR (fusion-model)	31.6	36
IMDAR (whole-framework)	70.6	18

Table 10
Trade-off between efficiency (seconds) and accuracy (%) of IMDAR on the NTU-RGB+D 60 dataset for C-Sub benchmark. The total time is calculated for a single skeleton sequence, including descriptors generation, initial predictions, and VPSS.

	Descriptors generation	Inital predictions	VPSS	Total Time	Accuracy
whole-framework	5.2×10^{-2}	2.2×10^{-4}	6.2×10^{-3}	$\approx 5.8 \times 10^{-2}$	92.8
fusion-model	5.2×10^{-2}	2.2×10^{-4}	-	$\approx 5.2 \times 10^{-2}$	85.2
two-descriptors (FDD+GD)	2.1×10^{-2}	7.3×10^{-5}	2×10^{-3}	$\approx 2.3 \times 10^{-2}$	87.0
single-descriptor (FDD)	10^{-2}	3.6×10^{-5}	-	$\approx 10^{-2}$	81.1

inference time, the whole-framework has a similar inference speed to RA-GCN, which is slower than other methods but offers better accuracy (Table 1). In contrast, the single-descriptor model demonstrates a high inference speed of 108 sequences/second.

4.6. Trade-off between efficiency and accuracy

Table 10 presents a trade-off analysis between efficiency (seconds) and accuracy. We observe that the inference time is primarily affected by the descriptor generation, then the VPSS algorithm, and much less by the initial prediction. The fusion model shows a decrease of 7.6% in accuracy without a noticeable increase in efficiency. Including only one descriptor leads to the best efficiency with just 10^{-2}, but with the lowest accuracy (81.1%). However, using only two descriptors shows a smaller decrease of accuracy to 5%, with 3.5×10^{-3} gain in efficiency, which is more convenient for practical applications since it balances between accuracy and efficiency.

Based on the results in Table 6, we observe that a dataset with fewer actions such as N-UCLA, require either a fusion model or a single-descriptor model to achieve acceptable accuracy. In real-world applications that require recognizing a limited set of actions, such as fitness tracking (e.g., detecting squats, jumping, push-ups), sports analysis (e.g., tennis serve, bat swing), or elderly care (e.g., detecting sitting, standing, walking, or falling), using two descriptors can provide high accuracy with lower computational cost. In conclusion, the efficient deployment of IMDAR for real-world action recognition depends on factors like the number of actions, the required accuracy, and the computational resources of the target device. For simpler applications or environments with limited resources, optimizing the pipeline with fewer descriptors offers a practical balance between accuracy and efficiency.

4.7. Qualitative evaluation

The evaluation of IMDAR effectiveness on actions that exhibit high similarity to other actions is shown in Fig. 9. We present the predictions, ground truth, and the key frames of each sequence, using samples from the testing set of the NTU-RGB+D 60 dataset. The actions ‘reading’ and ‘writing’ (first pair) are correctly classified despite their high similarity in global movements, with small differences observed in frames 60 and 80. IMDAR also accurately classifies highly similar actions, such as the pair ‘hand waving’ and ‘make a selfie’, which can be distinguished only by arm movement. However, when actions demonstrate even greater similarity, the model sometimes misclassifies actions, such as ‘neck pain’ as ‘headache’ and ‘eat meal’ as ‘sneeze/cough’.

In Fig. 10, we analyze the behavior of IMDAR on unseen actions in real-world videos. The action predictions are generated using a model trained on the NTU-RGB+D 120 dataset, with input skeleton sequences obtained from a 3D pose estimation model applied to the RGB video frames. The model demonstrates good generalization for actions performed similarly to those in the training data. For example, actions like ‘Tennis serve’ and ‘Throw something’, shown in rows 1 and 3, respectively, are correctly predicted. However, for actions that are partially similar to the training data, the model exhibits partial correctness. For instance, the action ‘Throw something’ (row 2) is classified as ‘Shot at basket’, because both involve throwing. Similarly, the action ‘Weight lifting stand’ is classified as ‘Stand up’, as the ‘Stand up’ action is the most similar action in the training set. When actions are performed differently from those in the training dataset, the model tends to misclassify them. For example, the ‘Tennis serve’ actions in rows 5 and 6, performed differently from the ‘Tennis serve’ action in row 1, and they are mistakenly classified as ‘Stretching oneself’ and ‘Sneeze/Cough’, respectively.

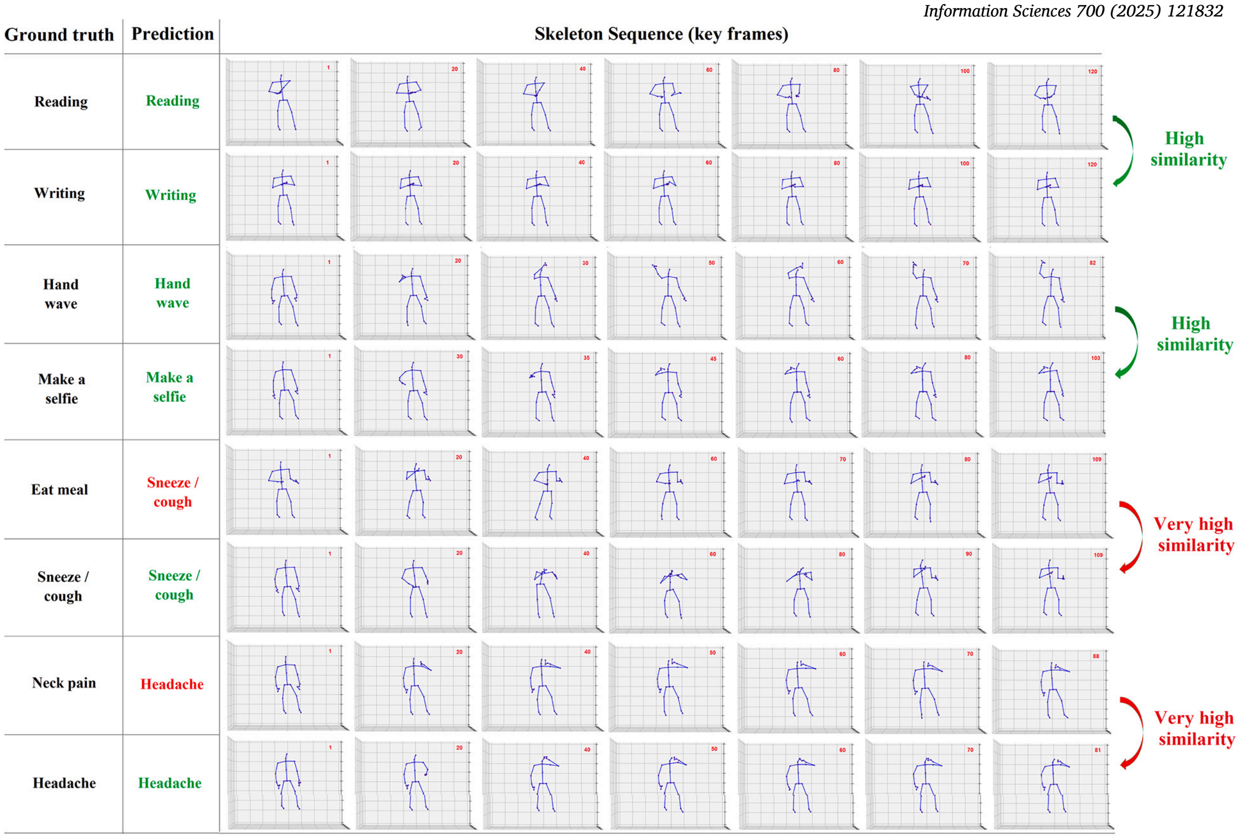

Fig. 9. Predictions alongside the ground truth for samples from the testing set of the NTU-RGB+D 60 dataset, where all test actions are captured with settings S = 001, camera view C = 001, performed by the performer P = 003, and trial R = 001. For each ground truth action, the key frames of the sequence are presented. Correctly classified actions are highlighted in green (despite high similarity with another action), while misclassified actions are highlighted in red (in cases of very high similarity with another action).

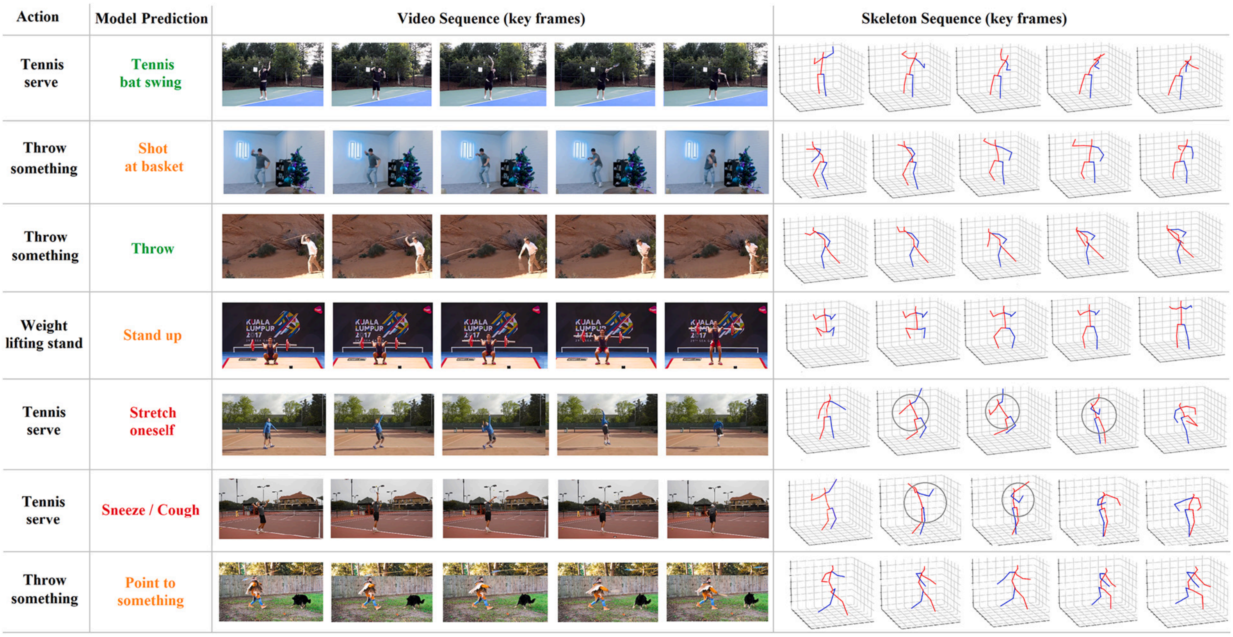

Fig. 10. IMDAR predictions on unseen actions of real-world videos. The predictions are obtained using the model trained on NTU-RGB+D 120 dataset. The skeleton sequence is generated by a 3D pose estimation model from the RGB video. Green: Correct prediction, Red: Incorrect prediction, Orange: Partially correct prediction. The circles indicate the pose similarity between the real action and the wrong prediction.

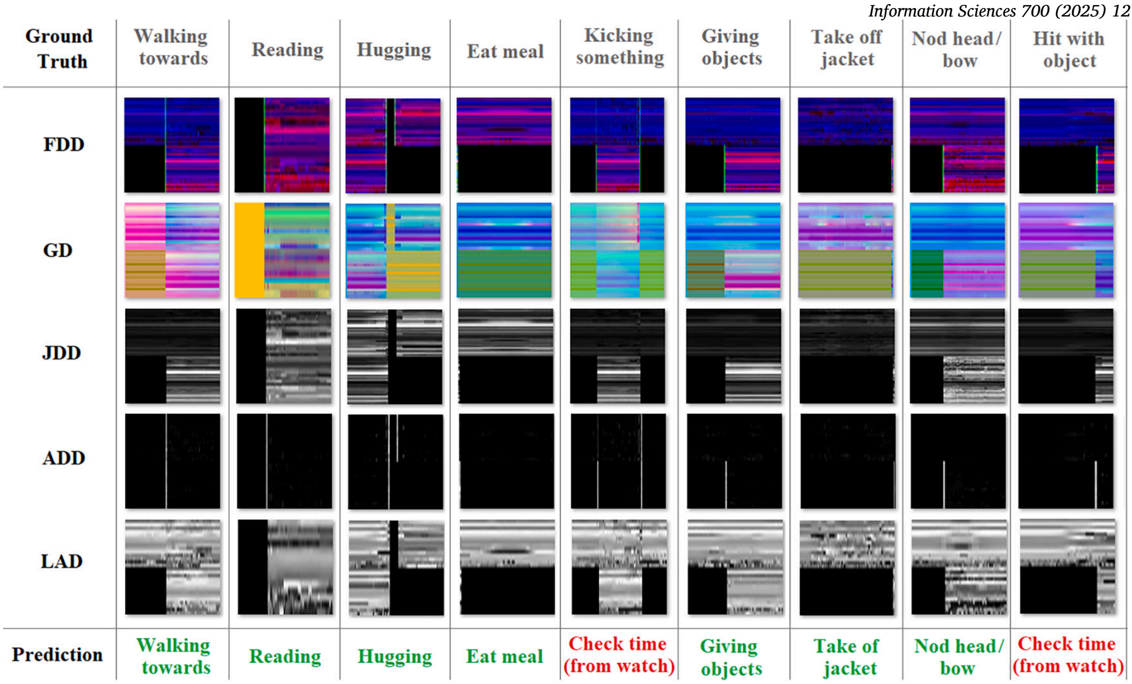

Fig. 11. IMDAR predictions on noisy samples from NTU-RGB+D 120 dataset. The black areas in the descriptors indicate missing skeletons or joints. The correct predictions are presented in green, and the wrong predictions are presented in red. For each action, the five descriptors are shown.

In the same way, the action ‘Throw something’ (last row) is misclassified as ‘Point to something’, likely due to the similarity in the arm movement.

To demonstrate the effectiveness of the diverse representations in handling missing skeletons or joints within a sequence, Fig. 11 presents several noisy samples, where the black areas indicate the missing skeletons or joints. In most cases, our method successfully classifies the actions, showing its robustness, even for interaction-based actions involving two individuals, such as ‘Walking towards’ and ‘Giving objects’.

4.8. Discussion

Advantages: IMDAR outperformed several state-of-the-art methods across four benchmark datasets with varying setups, camera view angles, action performances, and with different dataset sizes. The primary reason for this performance is the use of multiple representations of actions, which capture various aspects and critical features for holistic and complementary action representation through multiple descriptors combined with the VPSS algorithm. This was clearly demonstrated in the ablation study, where the absence of one descriptor impacts the overall accuracy (Table 6). Moreover, the analysis of the invariance capabilities of the descriptors indicates that accuracy is influenced more by how the action is performed than by pose, view angle, or sequence length (velocity). Furthermore, the qualitative evaluation of similar actions (Fig. 9), handling noisy data (Fig. 11), and the predictions of real-world actions (Fig. 10), demonstrate the robustness of IMDAR.

Limitations: Although IMDAR achieved high accuracy on multiple benchmarks, the analysis of the computation cost and inference time shows a high computation complexity compared to existing methods despite higher accuracy. The reason behind such complexity is the involving of six models for prediction, especially the fusion-model that include the five descriptors. Moreover, IMDAR shows a slower inference time, starting from the input skeleton sequence to the final prediction. This slow inference is mainly because of the constructions of five descriptors. Moreover, the VPSS algorithm investigates several options based on multiple predictions to decide the correct class, which influences the inference time.

Another limitation is the confusion between actions that have similar global and local skeleton movements, such as “eating meal” and “sneezing/coughing” (Fig. 9). In these cases, the lack of visual information (e.g., the presence of food) makes distinguishing them challenging, which require very precise local discriminative skeleton features.

Future work: Our proposed action representations have proven that they are crucial for efficient action recognition. However, using six models separately in parallel during the inference can be improved by stacking the five descriptors into a single representation with multiple channels, where each channel represents one descriptor. This approach allows to use a single model to process the stacked representations. But directly applying a single model to the stacked channels may not lead to the same performance as separated models, since features in some channels can influence those in others. Therefore, a careful design of the single model is essential. One solution for model design is to use 3D convolution to capture local features from multiple channels, and incorporate a self-attention mechanism to capture relationships between local features within each channel and global features across channels. The stacked representation will significantly reduce the computational time required, by eliminating the need for saving five descriptors for each sample, which reduces data loading by one-fifth. Additionally, using a well-designed single model that preserves the features of the five stacked descriptors will decrease the FLOPS by one-sixth.

To address the challenge of distinguishing between very similar actions that demonstrate local and global movements, a new proposed Motion Sensitive Descriptor (MSD) can be added to the stacked channels. This descriptor must focus exclusively on the most active joints and limbs, rather than including all joints, because the key factor in distinguishing actions in such cases is the body parts that exhibit more movement than the other parts.

5. Conclusion

In this paper, we introduced a new framework for skeleton-based action recognition, which consists of three modules; the action representation module, the feature extraction module, and the action prediction module. In the action representation module, we represented the skeleton sequence with five image descriptors, one descriptor represents the spatio-temporal relations change between joints over time, and four descriptors represent the distance change between joints and limbs over time. The descriptors are feature invariant, which means that similar actions are not affected by the pose, view angle, or velocity. In the feature extraction module, a well designed CNN model is used for feature extraction and classification of the five descriptors and their concatenated features. In the prediction module, six predicted classes are generated, and a VPSS algorithm was applied to select the correct class. Experiments have been conducted on four benchmark datasets. The comparison results and the ablation study demonstrate the effectiveness of our method.

CRediT authorship contribution statement

Kamel Aouaidjia: Writing – original draft, Software, Methodology, Conceptualization. **Chongsheng Zhang:** Writing – review & editing. **Ioannis Pitas:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

Kamel Aouaidjia and Chongsheng Zhang are supported in part by the National Natural Science Foundation of China (No. 62250410371) and the Henan Provincial Key R&D Project (No. 232102211021). Prof. Pitas was partially funded by the European Commission HORIZON Research and Innovation Actions under grant TEMA (No. 101093003, HORIZON-CL4-2022-DATA-01-01).

Data availability

Four public datasets used for evaluation

References

[1] A.F. Bavil, H. Damirchi, H.D. Taghirad, Action capsules: human skeleton action recognition, Comput. Vis. Image Underst. 233 (2023) 103722.

[2] C. Caetano, F. Brémond, W.R. Schwartz, Skeleton image representation for 3d action recognition based on tree structure and reference joints, in: 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), IEEE, 2019, pp. 16–23.

[3] C. Chen, R. Jafari, N. Kehtarnavaz, Utd-mhad: a multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor, in: 2015 IEEE International Conference on Image Processing (ICIP), IEEE, 2015, pp. 168–172.

[4] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, X. Sun, Measuring and relieving the over-smoothing problem for graph neural networks from the topological view, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 3438–3445.

[5] X. Chen, X. Luo, J. Weng, W. Luo, H. Li, Q. Tian, Multi-view gait image generation for cross-view gait recognition, IEEE Trans. Image Process. 30 (2021) 3041–3055.

[6] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, W. Hu, Channel-wise topology refinement graph convolution for skeleton-based action recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13359–13368.

[7] Z. Chen, S. Li, B. Yang, Q. Li, H. Liu, Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 1113–1122.

[8] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, H. Lu, Skeleton-based action recognition with shift graph convolutional network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 183–192.

[9] H.g. Chi, M.H. Ha, S. Chi, S.W. Lee, Q. Huang, K. Ramani, Infogcn: representation learning for human skeleton-based action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 20186–20196.

[10] A.M. De Boissiere, R. Noumeir, Infrared and 3d skeleton feature fusion for rgb-d action recognition, IEEE Access 8 (2020) 168297–168308.

[11] M. Ding, Y. Ding, L. Wei, Y. Xu, Y. Cao, Individual surveillance around parked aircraft at nighttime: thermal infrared vision-based human action recognition, IEEE Trans. Syst. Man Cybern. Syst. 53 (2022) 1084–1094.

[12] Y. Du, Y. Fu, L. Wang, Representation learning of temporal dynamics for skeleton-based action recognition, IEEE Trans. Image Process. 25 (2016) 3010–3022.

[13] G. Evangelidis, G. Singh, R. Horaud, Skeletal quads: human action recognition using joint quadruples, in: 2014 22nd International Conference on Pattern Recognition, IEEE, 2014, pp. 4513–4518.

[14] P.T. Hai, H.H. Kha, An efficient star skeleton extraction for human action recognition using hidden Markov models, in: 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE), IEEE, 2016, pp. 351–356.

[15] A. Hernandez Ruiz, L. Porzi, S. Rota Bulò, F. Moreno-Noguer, 3d cnns on distance matrices for human action recognition, in: Proceedings of the 25th ACM International Conference on Multimedia, 2017, pp. 1087–1095.

[16] Y. Hou, Z. Li, P. Wang, W. Li, Skeleton optical spectra-based action recognition using convolutional neural networks, IEEE Trans. Circuits Syst. Video Technol. 28 (2016) 807–811.

[17] B. Jin, X. Xu, Carbon emission allowance price forecasting for China Guangdong carbon emission exchange via the neural network, Glob. Finance Rev. 6 (2024) 3491.

[18] B. Jin, X. Xu, Forecasting wholesale prices of yellow corn through the Gaussian process regression, Neural Comput. Appl. 36 (2024) 8693–8710.

[19] B. Jin, X. Xu, Pre-owned housing price index forecasts using Gaussian process regressions, J. Model. Manag. (2024).

[20] N. Kilis, C. Papaioannidis, I. Mademlis, I. Pitas, An efficient framework for human action recognition based on graph convolutional networks, in: 2022 IEEE International Conference on Image Processing (ICIP 2022), 2022, pp. 1441–1445.

[21] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint, arXiv:1609.02907, 2016.

[22] C. Li, Y. Hou, P. Wang, W. Li, Joint distance maps based action recognition with convolutional neural networks, IEEE Signal Process. Lett. 24 (2017) 624–628.

[23] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, Q. Tian, Actional-structural graph convolutional networks for skeleton-based action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3595–3603.

[24] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.Y. Duan, A.C. Kot, Ntu rgb+ d 120: a large-scale benchmark for 3d human activity understanding, IEEE Trans. Pattern Anal. Mach. Intell. 42 (2019) 2684–2701.

[25] J. Liu, G. Wang, L.Y. Duan, K. Abdiyeva, A.C. Kot, Skeleton-based human action recognition with global context-aware attention lstm networks, IEEE Trans. Image Process. 27 (2017) 1586–1599.

[26] Y. Liu, H. Zhang, Y. Li, K. He, D. Xu, Skeleton-based human action recognition via large-kernel attention graph convolutional network, IEEE Trans. Vis. Comput. Graph. 29 (2023) 2575–2585.

[27] B. Nikpour, N. Armanfard, Spatial hard attention modeling via deep reinforcement learning for skeleton-based human activity recognition, IEEE Trans. Syst. Man Cybern. Syst. (2023).

[28] Y. Pang, Q. Ke, H. Rahmani, J. Bailey, J. Liu, Igformer: interaction graph transformer for skeleton-based human interaction recognition, in: European Conference on Computer Vision, Springer, 2022, pp. 605–622.

[29] C. Papaioannidis, I. Mademlis, I. Pitas, Fast cnn-based single-person 2d human pose estimation for autonomous systems, IEEE Trans. Circuits Syst. Video Technol. 33 (2023) 1262–1275.

[30] C. Plizzari, M. Cannici, M. Matteucci, Skeleton-based action recognition via spatial and temporal transformer networks, Comput. Vis. Image Underst. 208 (2021) 103219.

[31] A. Shahroudy, J. Liu, T.T. Ng, G. Wang, Ntu rgb+ d: a large scale dataset for 3d human activity analysis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1010–1019.

[32] L. Shi, Y. Zhang, J. Cheng, H. Lu, Two-stream adaptive graph convolutional networks for skeleton-based action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12026–12035.

[33] L. Shi, Y. Zhang, J. Cheng, H. Lu, Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition, in: Proceedings of the Asian Conference on Computer Vision, 2020.

[34] C. Si, W. Chen, W. Wang, L. Wang, T. Tan, An attention enhanced graph convolutional lstm network for skeleton-based action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1227–1236.

[35] C. Si, Y. Jing, W. Wang, L. Wang, T. Tan, Skeleton-based action recognition with spatial reasoning and temporal stack learning, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 103–118.

[36] Y.F. Song, Z. Zhang, C. Shan, L. Wang, Richly activated graph convolutional network for robust skeleton-based action recognition, IEEE Trans. Circuits Syst. Video Technol. 31 (2020) 1915–1925.

[37] Y.F. Song, Z. Zhang, C. Shan, L. Wang, Constructing stronger and faster baselines for skeleton-based action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 45 (2022) 1474–1488.

[38] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, Y. Li, Maxvit: multi-axis vision transformer, in: European Conference on Computer Vision, Springer, 2022, pp. 459–479.

[39] J. Wang, X. Nie, Y. Xia, Y. Wu, S.C. Zhu, Cross-view action modeling, learning and recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2649–2656.

[40] P. Wang, Z. Li, Y. Hou, W. Li, Action recognition based on joint trajectory maps using convolutional neural networks, in: Proceedings of the 24th ACM International Conference on Multimedia, 2016, pp. 102–106.

[41] A. Waswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: NIPS, 2017.

[42] Y. Wei, X. Chen, J. Gu, Human activity recognition soc for ar/vr with integrated neural sensing, ai classifier and chained infrared communication for multi-chip collaboration, in: 2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits), 2023, pp. 1–2.

[43] Y. Wu, X. Luo, Z. Xu, X. Guo, L. Ju, Z. Ge, W. Liao, J. Cai, Diversified and personalized multi-rater medical image segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 11470–11479.

[44] J. Xu, Y. Guo, Y. Peng, Finepose: fine-grained prompt-driven 3d human pose estimation via diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 561–570.

[45] K. Xu, F. Ye, Q. Zhong, D. Xie, Topology-aware convolutional neural network for efficient skeleton-based action recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 2866–2874.

[46] S. Yan, Y. Xiong, D. Lin, Spatial temporal graph convolutional networks for skeleton-based action recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018.

[47] F. Ye, S. Pu, Q. Zhong, C. Li, D. Xie, H. Tang, Dynamic gcn: context-enriched topology learning for skeleton-based action recognition, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 55–63.

[48] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, N. Zheng, View adaptive neural networks for high performance skeleton-based human action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2019) 1963–1978.

[49] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, J. Chen, Detrs beat yolos on real-time object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 16965–16974.

[50] H. Zhou, Q. Liu, Y. Wang, Learning discriminative representations for skeleton based action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 10608–10617.