Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Full length article

Transforming spatio-temporal self-attention using action embedding for skeleton-based action recognition☆

Tasweer Ahmad [a,*], Syed Tahir Hussain Rizvi [b], Neel Kanwal [b]

[a] *COMSATS University Islamabad, Sahiwal Campus, Pakistan*
[b] *Department of Electrical Engineering and Computer Science, University of Stavanger, Stavanger, Norway*

ARTICLE INFO

Keywords:
Human action recognition
Graph convolutional network
Link prediction
Self-attention
Computer vision

ABSTRACT

Over the past few years, skeleton-based action recognition has attracted great success because the skeleton data is immune to illumination variation, view-point variation, background clutter, scaling, and camera motion. However, effective modeling of the latent information of skeleton data is still a challenging problem. Therefore, in this paper, we propose a novel idea of action embedding with a self-attention Transformer network for skeleton-based action recognition. Our proposed technology mainly comprises of two modules as, (i) action embedding and (ii) self-attention Transformer. The action embedding encodes the relationship between corresponding body joints (e.g., joints of both hands move together for performing clapping action) and thus captures the spatial features of joints. Meanwhile, temporal features and dependencies of body joints are modeled using Transformer architecture. Our method works in a single-stream (end-to-end) fashion, where multiple-layer perceptron (MLP) is used for classification. We carry out an ablation study and evaluate the performance of our model on a small-scale SYSU-3D dataset and large-scale NTU-RGB+D and NTU-RGB+D 120 datasets where the results establish that our method performs better than other state-of-the-art architectures.

1. Introduction

Human action recognition has received a lot of attention for research now-a-days, thanks to the availability of publicly available multi-modal action datasets [1–3]. Recognizing actions in videos has numerous practical applications in surveillance, video content analysis, sports [4], health care, and entertainment etc. Considering the multi-modal data, convolutional neural network (CNN) [5,6] and graph convolutional network (GCN) [5,7] have shown remarkable performance for skeleton-based action recognition. Depth-sensors-based skeleton data is immune to illumination variation, background clutter, clothing, etc. [8]; therefore, contemporary methods extensively incorporate skeleton modality for action recognition.

The task of skeleton-based action recognition is challenging due to the scarcity of reliable spatially discriminative features and temporal dynamic models. Recently, spatio-temporal graph convolutional network (ST-GCN) has become a popular choice for skeleton-based action recognition that can efficiently represent the non-Euclidean data and effectively model the spatial and temporal dependencies [9–12]. However, ST-GCN faces some structural limitations. (i) The graphical representation of body skeleton is fixed for the input of a GCN, while body joints change their relative position while performing some action. Thus, a single fixed skeleton graph structure is not a suitable choice for all different action classes (e.g., wiping the face and brushing actions demand a stronger hand-to-head relationship, as compared to jumping and sitting down actions). (ii) The other drawback is that spatio-temporal convolution implemented as 2D convolution only utilizes limited local neighborhood information. To address such problems, we propose a new architecture by devising action embedding and a self-attention Transformer. The idea is based on graph embedding where an input graph is converted into a low-dimensional vector such that graph information is preserved [13], as shown in Fig. 1. Motivated by graph embedding, action embedding is studied to represent different human actions using low-dimensional feature vector. In this research, action embedding serves the purpose of modeling spatial features of body joints which could exploit the visual relationship between distant joints.

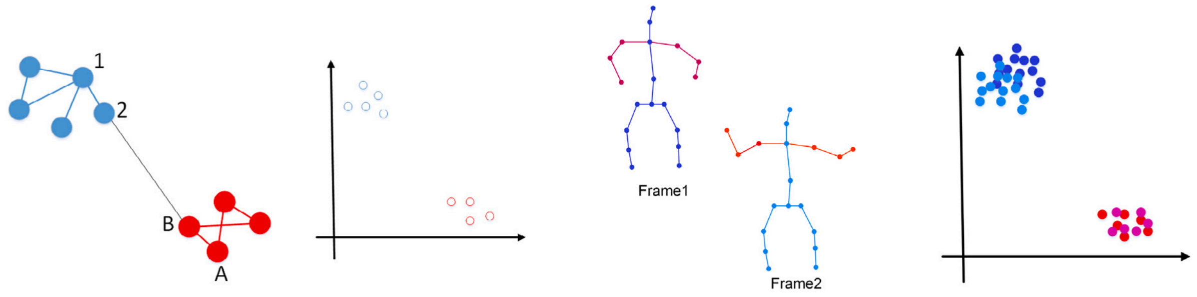
In our proposed architecture, action embedding is responsible for modeling spatial relationships, meanwhile, temporal features are modeled using Transformer architectures. Recently, Transformers have been shown to excellently model long-range temporal dependencies and thus have achieved superior performance in common image recognition tasks [14–16]. These factors motivated to employ self-attention

☆ This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author.

E-mail addresses: tasveerahmad@cuisahiwal.edu.pk (T. Ahmad), tahir.rizvi@uis.no (S.T.H. Rizvi), neel.kanwal@uis.no (N. Kanwal).

https://doi.org/10.1016/j.jvcir.2023.103892
Received 4 March 2023; Received in revised form 30 May 2023; Accepted 1 July 2023
Available online 12 July 2023



(a) Embedding of graph nodes in lower dimensional space

(b) Embedding of skeleton-graph nodes in lower dimensional space

Fig. 1. Illustration of graph embedding. On the left, the nodes with short edges in the graph stay close to each other in the embedded subspace. In the right figure, the nodes which are close to each other and moved in consecutive frames lie close to each other in embedded space.

Transformer to model the temporal relationship between body joints. Considering an example of a human skeleton performing some actions (e.g., clapping and running actions), distant body joints move concurrently to perform these actions. In the human body structure, the joints of each hand are at a distance from each other, and there is no bone connectivity between two hands. However, distant joints collaborate with each other while performing different actions. Therefore, an action relationship is established between such distant joints. This kind of action relationship is determined by action embedding using link prediction. In this work, we establish link prediction between distant joints by forming new edges which collectively participate in performing some action.

Our contribution for this work can be summarized as; 1. Action embedding is investigated for exploiting the spatial relationship between interacting joints 2. A self-attention Transformer network is proposed for modeling temporal dependencies between joints 3. A detailed ablation study is carried out for our proposed methodology for different observations. 4. Finally, the performance of our proposed method is compared for three skeleton-based action recognition datasets where experimental results show that our method performs better than other methods.

We organize the rest of our paper by discussing a comprehensive literature survey on *action-embedding, self-attention Transformer*, and *action recognition* in Section 2. Section 3 presents our proposed methodology. The dataset, experimental setup, and results are described in Section 4. Section 5 visualizes results, and finally, we conclude our work in Section 6.

2. Background

In the last decade, CNN and GCN architectures with different variants remained popular choices for action recognition [17,18]. In this section, we survey the literature and organize it further as follows; (i) Graph-Action Embedding (ii) Self-attention Transformer (iii) Skeleton-based action recognition.

2.1. Action embedding

Our motivation for action embedding comes from graph embedding, which can broadly be grouped into three main categories: factorization-based, random walk-based, and deep learning (DL) based methods [18,19]. Perozzi et al. [20] proposed a novel technique named DeepWalk for learning a latent representation of vertices in a graph network. DeepWalk included local information from random walks for learning social representations of vertices, such as neighborhood similarity and community membership as latent features. Borrowing the idea from sociology and linguistics, second-order proximity can be interpreted as nodes with shared neighbors, which are likely to be similar. Another approach by [21], which could embed millions and billions of nodes, is named LINE (Large-scale Information Network Embedding). This approach was efficient for embedding first-order and second-order proximity of graph. A graph embedding technique named node2vec, using biased random walk, was proposed by Grover et al. [22]; it explored the diverse neighboring nodes in a graph. This proposed method was efficacious for link prediction and multi-label classification on challenging datasets. In [22], authors applied node2vec for link prediction in Facebook dataset (having 4039 nodes and 88,234 edges), Protein–Protein Interactions dataset (having 19,706 nodes and 390,633 edges) and arXiv ASTRO-PH dataset (18,722 nodes and 198,110 edges). Going deeper for modeling the structural similarity in the graph, Ribeiro et al. [23] devised a struc2vec method for node similarity. The struc2vec achieved excellent performance for node classification tasks which are strongly dependent on structural identity.

A new concept of task-independent graph representation algorithm, known as anonymous walk, is recently introduced by [24]. Adopting the concept of isomorphic graphs, invariant graph embedding (IGE) proposed by [25] relies on spectral graph theory. The authors claim that their approach is a powerful feature representation technique for a large family of graphs.

2.2. Transformers in action recognition

An abridged version of spatio-temporal Transformer network was presented by [26] and later extended by [27]. This methodology works by devising two-stream architecture as spatial self-attention and temporal self-attention stream. The spatial self-attention stream extracts intra-frame interaction between body parts, whereas temporal self-attention highlights the inter-frame correlation. This method is implemented in a two-stream fashion, which adds complexities to its end-to-end implementation pipeline. The work proposed by [26] includes action embedding as input for a single-stream Transformer network. Action Transformers proposed by [28] recognize and localize human actions in videos using ConvNets as I3D. This algorithm is powerful in learning the semantic context around performer.

Video Transformer Network (VTN) mitigates the computational burden of 3D ConvNet by incorporating 2D ConvNet for spatial feature extraction and a self-attention Transformer for temporal feature extraction [29]. However, video transformer network (VTN) puts a restriction on using only raw positional encoding with vanilla-Transformer architecture. In [30], Action-centric Transformer models action-based encoding of frames, while Relation-Transformer establishes the temporal relationship between frames. Except [26], all methods cited above are excited with RGB as input frames for the Transformer network.

2.3. Skeleton-based action recognition

For skeleton-based action recognition, the skeleton (body joints) positions are used as the input features [17,31]. We broadly divide our literature on skeleton-based action recognition into the following two categories.

CNN-RNN Techniques: The study accomplished by [31,32] investigates the trust-gate long short-term memory networks (LSTM) for modeling spatial and temporal features for skeleton-based action recognition. However, they did not take into account the longer spatio-temporal dependency. Later [33] proposed Skeleton-net, which extracts features from the sequence of body-skeleton and transforms them into images that are fed to CNN for classifying actions. The idea of clipping representation using the 3D coordinate sequence was proposed in [34]. Their approach used multitask CNN for action classification. Another research has incorporated multimodal data for pose estimation [35,36]. Hong et al. [36] developed a deep autoencoder-based method for human pose recovery. Their technique employed a multimodal deep autoencoder to learn a joint representation of the 2D image and 3D pose data, which can extract the 3D pose from a single 2D image. Later, in another work, the authors [35] proposed the multitask manifold deep learning (M^2DL) framework to estimate facial posture. Their technique enhanced the multimodal mapping relationship with multitask learning that helps to estimate the gazing direction or head postures with 2D images. For recognizing actions, Ou et al. [37] suggest a 3D deformable convolution temporal reasoning (DCTR) network. They employed Conv-LSTM to predict the long-term temporal dynamics of activities and 3D deformable convolution to capture the spatio-temporal information of the input RGB video. Yu et al. [38] presented the hierarchical deep word embedding (HDWE) model to recognize click features in RGB images. Their approach used a coarse-to-fine click feature predictor that was trained using an additional picture dataset containing click information.

Some of the human actions may involve multi-person interaction (e.g., hugging, pointing fingers, etc.); therefore, Shu et al. [39] proposed concurrent-LSTM for modeling such human actions and interactions. The authors introduced a separate LSTM to encode the static features of each person, which are fed to concurrent LSTM responsible for integrating and storing inter-related motion information of several interacting persons via the cell gate. Recent RNN techniques for skeleton-based action recognition lack the spatial coherence between joints and temporal evolution of body-skeleton. In [40], authors proposed skeleton-joint co-attention RNN for figuring out the spatial coherence of joints and temporal evolution among skeletons. Our work overcomes the challenges exhibited by CNN and RNN, by utilizing longer spatio-temporal multi-head attention, which can be prolonged to 300-frames.

GCN Techniques: Due to the close resemblance of body-skeleton to a graph, a number of researchers are applying GCN for skeleton-based action recognition [41–43]. The pioneering work for skeleton-based spatial–temporal action recognition was carried out by [42]. In this method, human body skeleton was represented as a spatial–temporal graph which is then excited to a GCN. This method shows the best performance for NTU-RGB+D and Kinetics-Skeleton datasets. However, the major limitation of their work is the fixed skeleton-graph structure. Shi et al. [9] addressed the problem of fixed skeleton-graph by devising two-stream adaptive GCN where the adjacency matrix is constrained to be learnable from the training data. The authors used bone-stream and joints-stream as two streams of GCN network. This is also regarded as pioneer work for skeleton-based action recognition using GCN. Zhao et al. [43] devised a method using structure-aware feature representation by incorporating LSTM. Their approach modeled spatial dependency using Bayesian neural network and temporal dependency using LSTM. Later, Shi et al. [10] proposed the idea of directed GCN (D-GCN) for skeleton-based action recognition. In this method, human skeleton was represented as a directed acyclic graph, where edges were directed from the center of the body toward outside. D-GCN was trained as a separate two-stream network and, therefore, undergoes the limitation of doing end-to-end training. Ahmad et al. [44] proposed GCN-based action recognition using graph sparsification by edge effective resistance. This work mainly investigated attention joints for GCN-based action recognition.

Peng et al. [45] introduced an idea of neural architecture search (NAS) and presented the first automatically designed GCN for skeleton-based action recognition. The authors evaluated their method on NTU-RGB+D and Kinetics-skeleton dataset, where it showed improved performance. In a different approach, Chen et al. [46] developed structure-based graph pooling (SGP) with joint-wise channel attention (JCA) for skeleton based GCN. SGP captures more global representation and mitigates the parameters for computation. However, SGP has a limitation that it is computed using hand-crafted methods by dividing the original graph into sub-graphs. Liu et al. [11] proposed a multi-scale aggregation scheme for disentangling important nodes in the neighborhood. In their method, skip connections were introduced as dense cross-space–time edges, and the method was evaluated on NTU RGB+D and Kinetics Skeleton datasets. Recently, Liu et al. [47] combined GCN with the hidden conditional random field for retaining the structural information of the human skeleton. Their proposed method was trained in an end-to-end manner and then tested on NTU-RGB+D, N-UCLA, and SYSU datasets. A combination of Hierarchical spatial reasoning (HSR) and temporal stack learning network is introduced by [5]. HSR captures the two levels of information (i) Intra-spatial information of each body part, and (ii) Body-level information between parts. Temporal stack learning is responsible for modeling temporal information. For capturing the spatial and temporal dependencies in the skeleton sequences, [48] integrated GCN and LSTM networks. This methodology introduced an attention mechanism that focuses on discriminative joints and thus enhances the model's ability to pay attention to the relevant information. Cho et al. [49] proposed a DL model for recognizing human actions from skeleton data using a self-attention mechanism. The authors employed self-attention to capture the most important joints and temporal relations among them. Their newly proposed normalization technique, referred to as "skeleton normalization", enhanced the robustness of the model to variations in scale and rotation. Using skeleton data, [50] presented a DL framework for identifying human actions. To account for both spatial and temporal dependencies in the skeletal sequences, the model used spatio-temporal attention mechanisms in such a manner that it could efficiently focus on discriminative joints and their temporal dynamics by including attention mechanisms at various levels. Our proposed end-to-end architecture eliminates the requirement for handcrafted feature engineering and delivers competitive performance on benchmark datasets.

3. Methodology

In this section, we discuss our methodology by proposing a novel idea of action embedding for recognizing different actions using self-attention Transformer. Our proposed architecture consists of three main parts;

(i) Spatial features are extracted by applying action embedding on the input skeleton data.
(ii) Self-attention Transformer models the temporal features of the input data.
(iii) Multiple-layer perceptron is excited with spatial and temporal data for classification.

The pipeline of our proposed methodology is explained in Fig. 2, where action embedding features are first extracted using graph embedding techniques, such as DeepWalk or Graph Convolution, etc. Then on top of action embedding, link prediction is carried out in order to establish the relationship between distant joints. Action embedding is signified by representing spatial features into lower-dimensional flattened vector. In the next stage, a flattened feature vector is fed to encoder–decoder self-attention Transformer which exploits the temporal contextual information and thus models the inter-dependencies between joints. Finally, the output of Transformer network is fed to MLP for action classification.

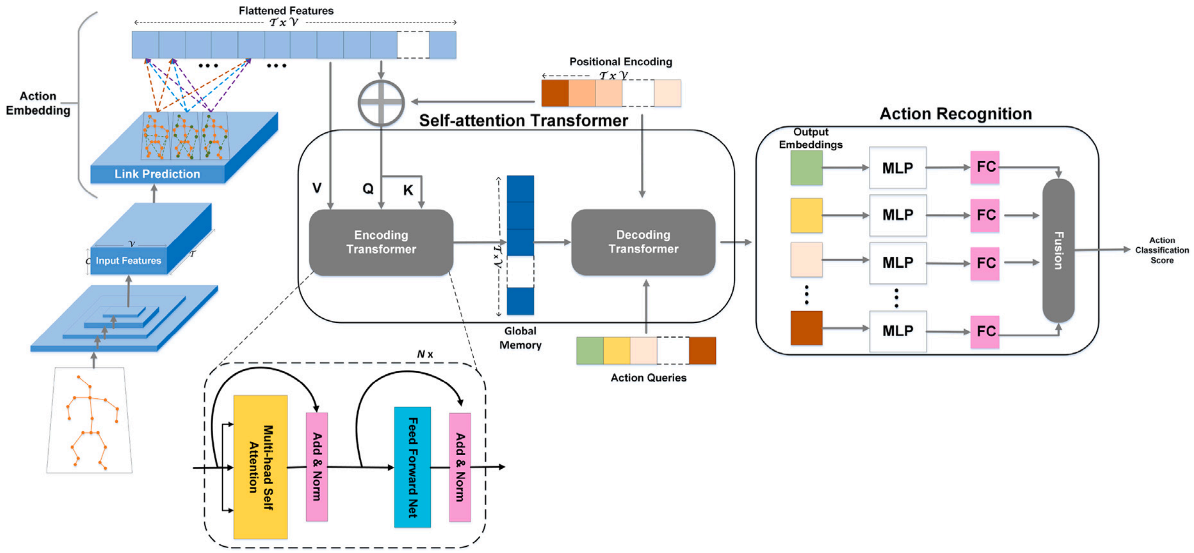


Fig. 2. Block diagram of our proposed method for recognizing actions using action embedding and Self-attention Transformer. Action embedding predicts the relationship between distant body joints for performing some action. Self-attention Transformer extracts temporal inter-frame dependencies for joints. The output embeddings are fed to MLP for final action classification.

3.1. Action embedding

We obtain the action representation using graph embedding, which is used for representing graph nodes in a lower-dimensional space, such that similarity or relationship between neighboring nodes is preserved. Graph embedding is implemented by using,

(i) Random walks
(ii) Graph factorization
(iii) Node-proximity in the graph

Random walks can approximate a number of graph properties which can be the node centrality [51] and node similarity [52]. Random walks can be implemented by using node2vec [22], Deepwalk [20], and struc2vec [23]. Since the random walk method can represent graph information; therefore, if we can control the walks on a graph then we can also manipulate the information that can be embedded in a given graph.

Action embedding, z_v, for a node v is defined in terms of mapping function, f_{map}, which maps the nodes of a graph into lower d-dimensional space, by following the relationship,

$$z_v = f_{map}(v) \in [0,1]^{n \times n \times C} \tag{1}$$

f_{map} is a matrix of size $|V| \times d$ parameters; meanwhile, the notation C corresponds to the different contexts in which links may be combined; for example, joints-velocity and orientation, etc. It is formulated that if distant joints are moving relatively at the same velocity and have the same trajectories, then some sort of relationship can be established between those distant joints. A similarity between distant nodes u and v is defined in terms of encoding function as,

$$\text{similarity}(u, v) \approx z_v^T z_u \tag{2}$$

This similarity function is based on Euclidean distance (dot-product) between nodes of a graph in encoding space. The neighborhood feature aggregation from level 'l' to '$l+1$' is defined as,

$$\mathrm{h}_v^l = \sigma(\mathrm{W}_l \sum_{u \in N(v)} \frac{h_u^{l-1}}{|\mathrm{N}(v)|} + B_l h_v^{l-1}) \quad \forall \quad l \in 1, \ldots, L \tag{3}$$

The symbol σ denotes the non-linearity, W_l, and B_l are the learnable weight and bias matrices. In order to compute the representation of a node v at level l, it is required to aggregate the neighbors of this node in the previous level $l-1$. The factor $\sum_{u \in N(v)} \mathrm{h}_u^{l-1}/|\mathrm{N}(v)|$ corresponds to aggregated response in the previous level.

Intuitively, we combine the information from neighbors of a node v in the previous level, then the information of node v itself is combined for graph embedding. Elaborating the concept of graph embedding, which works hierarchically, for node v, the information of its neighbors is combined at level l, then the information of neighbors-of-neighbors is combined at level $l+1$, and so on. In Eq. (3), if W_l is set to zero, then h_v^l is not learning from its neighbors and only incorporating the information from itself. Contrary to it, if W_l is set very high, then features of node v itself are substantially ignored, and information is borrowed from neighbors. Therefore, it becomes an optimization problem that the amount of information being accumulated from the neighbors of node v and the information from the node itself being contributed for prediction.

We apply vector concatenation across different levels for action embedding using vector representations of node v as,

$$\mathbb{Z}_v = [h_v^1, h_v^2, \ldots, h_v^l], \quad \forall \quad v \in \mathbb{V} \tag{4}$$

where $\mathbb{Z}_v$ is the final action embedding after k-levels of the neighborhood aggregation, which is followed by a single fully-connected softmax layer for link prediction between vertices as,

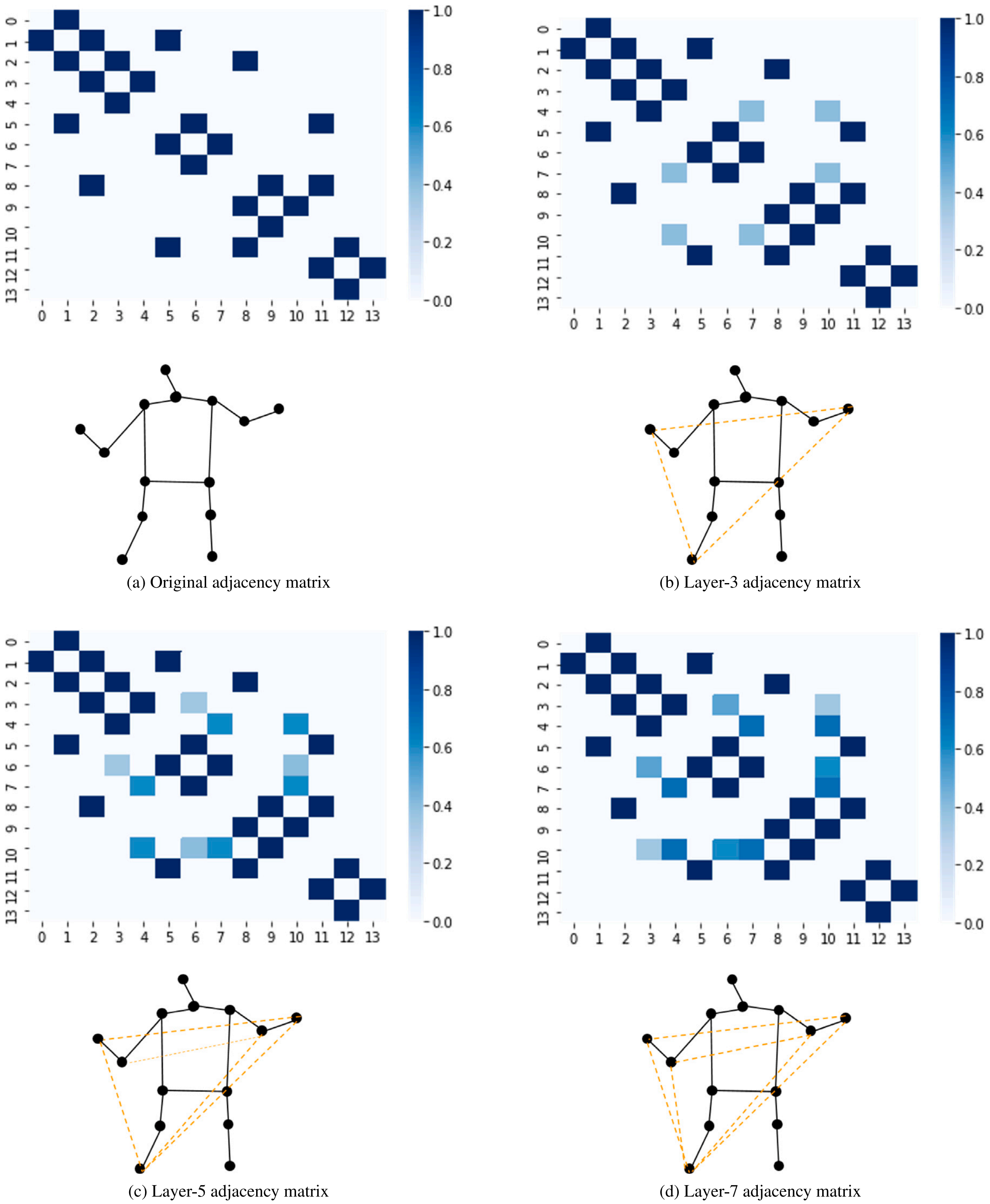
$$\hat{y} = \mathrm{softmax}(\mathbf{W} \times \mathbb{Z}_v + \mathbf{B}) \tag{5}$$

where $\hat{y}$ denotes the output probability, whether there exists a link between nodes or not.

Using action embedding, the adjacency matrix learns new edges and relationships between nodes of a skeleton graph. The adjacency matrix gradually learns in a layer-wise fashion, while traversing different layers, some new action edges are established between nodes, and some existing action edges are strengthened. This concept is further illustrated in Fig. 3.

3.1.1. Link prediction for action recognition

Graph embeddings are suitable to represent a graph with lower-dimensional vectors and matrices. Link prediction is one of the major applications of graph embedding for generating new edges between structurally similar nodes. The link prediction approaches include



(a) Original adjacency matrix

(b) Layer-3 adjacency matrix

(c) Layer-5 adjacency matrix

(d) Layer-7 adjacency matrix

Fig. 3. Adjacency matrix learns new action edges between joints using action embedding. By increasing the number of layers, new edges are learned as action links, and existing links are also strengthened gradually.

similarity-based methods, maximum-likelihood models, and probabilistic models [19]. Graph embedding implicitly captures the inherent dynamics in a graph network and thus enables link prediction based on this inherent relationship. Our motivation for using link prediction comes from the concept that distant joints and limbs move concurrently despite no physical connection between them. For example in running action, two legs and their corresponding joints move together, therefore, a relationship (a possible edge) should be established between the knees and ankles of the two legs. We understand this concept and try to model such a relationship between distant joints using link prediction by graph embedding. Feature learning for link prediction is made on the following two assumptions:

Conditional Independence: Following the neighborhood sampling strategy, $N_S(u) \subset V$ is defined as the neighborhood of node u. The likelihood of observing a neighborhood node is factorized by the assumption that likelihood is independent of observing any other neighboring node:

$$P_r(N_S(u)|f_{map}(u)) = \prod_{n_i \in N_S(u)} P_r(n_i|f_{map}(u)) \tag{6}$$

Feature Space Symmetry: It is built on the assumption that there exists a strong symmetric effect over the source and neighboring nodes in the feature space. This symmetry is modeled as the conditional likelihood of every source-neighborhood node pair as a softmax function, defined as a dot product of their features.

$$\mathrm{P_r(n_i|f_{map}(u))} = \frac{exp(f_{map}(n_i).f_{map}(u))}{\sum_{v\in V} exp(f_{map}(v).f_{map}(u))} \tag{7}$$

In link prediction, the neighborhoods of a node are examined and sampled using local search methods. Borrowing the idea from sociology, there exists two types of equivalences (i) Homophily equivalence corresponds to the nodes which are of a similar type and close to each other, and (ii) Structural equivalence where nodes may exist at distance in a graph structure. BFS corresponds to homophilous nodes and generates a microscopic view of the graph structure, whereas DFS figures out macroscopic details in a graph, explained in Fig. 4. In random walks for link prediction, node traversing is determined by transition probability among the nodes. In this regard, weights of edges play an important role in computing the transition probability [22]. Following the taxonomy of [19], Deepwalk and node2vec are two very important random walk methods.

DeepWalk Method: In literature, random walks have been used as a similarity measure for numerous applications such as community detection [53], link prediction and content recommendation [52]. A random walk rooted at vertex v_i is denoted as $\mathcal{W}_{v_i}$, is modeled as a stochastic process with random variables $\{\mathcal{W}^1_{v_i}, \mathcal{W}^2_{v_i}, \ldots, \mathcal{W}^K_{v_i}\}$. Borrowing the concept from language modeling [20] and representing body joints as vertices, $\{v_1, v_2, \ldots, v_{i-1}\}$, the likelihood of estimating joint-occurrence for a particular action is,

$$\mathrm{P_r}(v_i|(v_1, v_2, \ldots, v_{i-1})) \tag{8}$$

An intuitive understanding of this concept is built to estimate the likelihood of co-movement for vertex v_i, given all the previous vertices visited so far. This concept of co-movement is introduced as mapping function $f_{map} : v \in V \rightarrow \mathbb{R}^{|V|\times d}$. Using the above expression, likelihood estimation is re-written in the form of a latent representation,

$$\mathrm{P_r}(v_i|(f_{map}(v_1), f_{map}(v_2), \ldots, f_{map}(v_{i-1}))) \tag{9}$$

The corresponding objective function for the optimization is formulated as,

$$min_{f_{map}} - log\left\{\mathrm{P}_r\left(v_{i-w}, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{i+w}|\left(f_{map}\left(v_i\right)\right)\right)\right\} \tag{10}$$

for DeepWalk as action embedding, Eq. (10) is solved which models the motion similarity between neighboring vertices of a graph.

Node2vec Method: For node2vec, we consider random walks along edge E originating from a random node $v_0 \in V$ by repeatedly sampling an edge to transition to the next node $v_{i+1} := sample(E[v_i])$, where $E[v_i]$ represents the outgoing edges from v_i. When a node2vec algorithm is applied with transition sequences $\{v_0 \rightarrow v_1 \rightarrow v_2 \ldots\}$, it learns the embeddings by stochastically considering every node along the sequence, v_i. Thus for this *anchor* node, v_i, the embedding representation is brought closer to the embedding of its neighbors $\{v_{i+1}, v_{i+2}, \ldots, v_{i+c}\}$, the *context nodes*. Generally, the context window size is sampled from the uniform distribution $U\{1, C\}$, as explained in [54]. Considering a random walk having co-occurrence matrix $D \in \mathbb{R}^{|V|\times|V|}$ where each entry D_{vu} in the matrix corresponds to the number of times nodes v and u are moved together by some orientation within a context distance $c \sim U\{1, C\}$ in all random walks. Using node-2-vec as the underlying architecture for action embedding, the loss function can be optimized using the negative log-likelihood of softmax,

$$\min_y[logZ - \sum_{v,u\epsilon V} D_{v,u}(Y_v^T Y_u)] \tag{11}$$

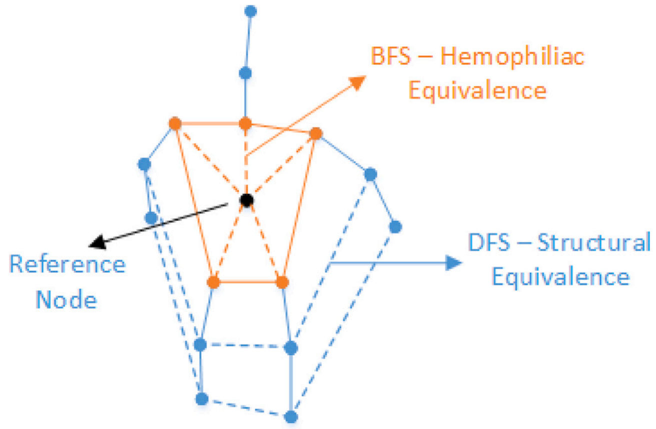


Fig. 4. An example of difference between BFS and DFS.

The partition function is estimated by using negative sampling, $Z = \sum_{v,u} exp(\hat{Y}_v^T \hat{Y}_u)$. Node2vec traversal procedure is based on two parameters as return parameter and in-out parameter [22].

Graph Convolution: Action embedding is also implemented using a graph convolution architecture [55]. In this method, node features denote spatial locations of body joints and the displacement of joints in successive frames. The input feature map $C \times \tau \times V$ corresponds to the number of channels C, frame-length τ, and body-skeleton vertices V. From hidden layer H^l to H^{l+1}, convolution operation for a graph $G = (V, E)$ with vertices $|V|$ and edges $|E|$ is defined as,

$$H^{l+1} = \widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}H^l W^l \tag{12}$$

where W^l represents the corresponding layer-specific linear transformation's weight matrix, H^l is the feature map activation in the lth layer such that $H^0 = X$, whilst H^{l+1} denotes the updated hidden layer feature matrix. The adjacency matrix is defined as $\widetilde{A} = A + I_N$, while degree matrix is represented as $\widetilde{D_{ii}} = \sum_j \widetilde{A_{ij}}$. For GCN, the embedding function is written as,

$$Z_{action} = \mathrm{softmax}(\widetilde{A} \cdot \mathrm{ReLU}(\widetilde{A}X^t H^t) \cdot H^{t+1}) \tag{13}$$

The softmax function in the above expression is calculated as $softmax(x_i) = \frac{exp(x_i)}{\sum_j exp(x_j)}$ in a row-wise fashion.

3.2. Self-attention transformer network

Self-attention Transformers are shown to be effective architectures for modeling temporal sequences in various domains [26,56]. In a broader context, self-attention Transformers comprise of encoding and decoding blocks.

Encoder: Self-attention encoding block is built using multi-headed self-attention (MSA) layer feed-forward network (FFN) with residual skip-connections after every block, [57,58]. In MSA, the information of different subspaces at various positions is concatenated. Compared to LSTM and recurrent networks, self-attention networks are better at learning long-range dependencies [14]. FFN is a two-layer network with gaussian error linear unit (GELU) as non-linearity [59]. Since Transformer architecture is permutation invariant, position encoding [60,61] is also added as an input to each attention layer. The flattened feature and positional encoding are aggregated to feed the Transformer encoder for summarizing global information. We denote the output of the encoding block as global memory, shown in Fig. 2. Our Transformer network receives a 1-D sequence of action embedding as input. We represent the input skeleton-graph as $x \in \mathbb{R}^{V\times E\times C}$, where V denotes the number of vertices, E represents the number of edges and C corresponds to the number of channels. Through all layers, Transformer network uses a constant latent vector of size D which maps the input vector x to D dimensions using trainable linear projection, as referred in Eq. (14). With the action embedded sequences ($z_0 = x_{class}$), we append learnable positional embedding whose output at Transformer encoder serves as image representation z_{enc}, as shown in Eq. (16). We use one-dimensional positional embedding along with action embedding which is fed to the encoder input. Referring to Eqs. (15) and (16), layer normalization is applied before MSA and FFN blocks.

$$\begin{aligned} \mathbf{z}_0 = [x_{class}; x^1 \mathbf{E}_{ae}; x^2 \mathbf{E}_{ae}; \dots ; x^N \mathbf{E}_{ae}] + \mathbf{E}_{pos}, \\ \mathbf{E}_{ae} \in \mathbb{R}^{N\times D}, \mathbf{E}_{pos} \in \mathbb{R}^{N\times D} \end{aligned} \tag{14}$$

$$\mathbf{z}'_l = \mathrm{MSA}(LN(\mathbf{z}_{l-1})) + \mathbf{z}_{l-1}, \quad l = 1, \dots, L \tag{15}$$

$$\mathbf{z}_{enc} = \mathrm{FFN}(LN(\mathbf{z}'_l)) + \mathbf{z}'_l, \quad l = 1, \dots, L \tag{16}$$

In Eq. (14), we apply action embedding E_{ae} to the input skeleton graph, which is then concatenated with positional embedding and excited to MSA in Eq. (15). A FFN is excited with the output of MSA along with residual connection in Eq. (16) for generating encoded output.

Decoder: We implemented a decoding block using a stack of N identical layers of MSA and FFN. The decoder network includes an additional block of multi-head attention as a cross-attention layer from the output of the encoder. The decoder transforms N learned positional embeddings (see action queries in Fig. 2) into N output embeddings. Similar to the encoder, each sub-layer of the decoder is employed with layer normalization and has three inputs as (i) Global memory from the encoder, (ii) Action-queries, and (iii) Positional encodings. For decoding multi-head cross-attention, the values are directly provided from global memory. Global memory and positional encoding are summed as key vectors; whilst query vector is the summation of input positional encoding and input action queries. The self-attention layer is implemented by providing query, key, and value from action queries or the output of the decoder layer. We denote the output of the decoder as output embedding. Our encoder–decoder transformer layer implementation is similar to work [62].

$$\mathbf{z}''_l = \mathrm{MSA}(\mathrm{LN}(\mathbf{z}_{enc})) + \mathbf{z}_{enc}, \quad l = 1, \dots, L \tag{17}$$

$$\mathbf{z}_{dec} = \mathrm{FFN}(\mathrm{LN}(\mathbf{z}''_l)) + \mathbf{z}''_l, \quad l = 1, \dots, L \tag{18}$$

3.3. Action recognition

The output embeddings from the Transformer network are then fed to the MLP layer for final classification. The classifier provides the state of action, as shown in Eq. (19).

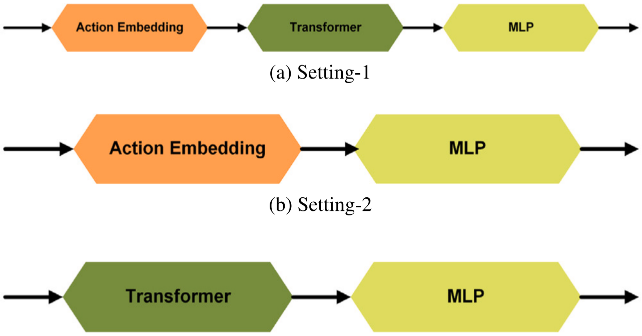
$$\mathbf{y} = \mathrm{MLP}(\mathbf{z}^0_{dec}) \tag{19}$$

4. Experiment setup and results

In this section, we provide an overview of the dataset, experimental details, and results.

4.1. Datasets

SYSU-3D: Hu et al. [1] prepared a human activity SYSU-3D dataset, having 480 videos of 40 different subjects engaging in 12 different activities. The dataset contains 30 coordinates from 20 different joints associated with each frame of the sequence. There are two settings available for this dataset; For both settings, half of the video samples are used for training while the remaining half is used for testing purposes. **NTU-RGB+D:** NTU-RGB+D dataset [2] contains 60 different action classes and 56,578 skeleton sequences that are collected from 40 different subjects and 3 different camera viewpoints. For the dataset, each body skeleton is represented by 25 distinct joints. The classification accuracy on this dataset is reported for two settings as (i) cross-subject (X-Sub), and (ii) cross-view (X-View). This dataset has been extensively used for evaluating the models for large-scale skeleton-based action recognition.



(a) Setting-1

(b) Setting-2

(c) Setting-3

Fig. 5. Different experimental settings used for ablation study.

NTU-RGB+D-120: NTU-RGB+D-120 dataset [3] is an extension of NTU-RGB+D, which contains 120 different action classes. There are 113,945 skeleton sequences in this dataset, which have been collected from 106 different subjects. Similar to the previous dataset, we have tested two evaluation settings for this dataset (i) cross-subject (X-Sub) (ii) cross-setting (X-Set).

4.2. Implementation

Our experimental setup was based on Pytorch [63] deep learning framework. We train our models for 200-epochs using a batch size of 32. An initial learning rate was set to 0.1, with learning rate decay. We used ADAM as the training optimizer, and each experiment was carried out three times for cross-validation. During experimentation, we used 8-heads for multi-head attention. All models were trained on Tesla T4 16 GB GPU with compute unified device architecture (CUDA) version 11.0. It took roughly 8-hours to train the models under our experimental setup.

4.3. Ablation study

We carried out an ablation study by investigating the efficacy of our proposed model, action embedding techniques, and Transformer depth. Our ablation study used a five-layer GCN as the backbone architecture for extracting features from skeleton data. We trained our model for 200 epoch with one-time learning rate decay at epoch 150. For the proposed study, Transformer architecture employed four encoding and decoding layers with a batch size of 32.

4.3.1. Efficacy of action-embedding transformer

We deployed three different settings as shown in Fig. 5, then investigated the effect of action embedding and self-attention network in each setting. For setting-1, all three blocks were included, which marks the highest accuracy by endorsing the complementarity of these blocks. Meanwhile, setting-2 included action-embedding and MLP, which established the significance of spatial attention. The classes that benefited most from action-embedding are “clapping hands” and “running”, where distant joints contribute to some actions. Setting-3 incorporated Transformer and MLP blocks, highlighting the importance of temporal attention. “Hugging”, “handshaking”, and “pushing other people”

Table 1
Performance comparison for different settings. Best accuracy results are marked in bold.

	Setting-1	Setting-2	Setting-3
NTU-RGB+D	**84.4**	82.6	82.0

Table 2
Performance comparison for varying the depth of Transformer. Best accuracy results are marked in bold.

Number of layers	NTU	NTU-120
1	86.5	84.1
2	87.0	84.6
4	87.6	85.0
8	**88.3**	**85.7**

Table 3
Performance comparison for different link prediction methods. Best accuracy results are marked in bold.

Dataset	Complexity	SYSU-3D	NTU	NTU-120
DeepWalk	O($\|V\|d$)	78.9	75.6	73.3
Node2vec	O($\|V\|d$)	83.5	80.4	77.8
GCN	O($\|V\|d^2$)	**88.4**	**85.8**	**84.2**

Table 4
Performance comparison with the state-of-the-art architectures for SYSU-3D dataset. Best accuracy results are marked in bold.

Method	SYSU-3D	Year
ST-LSTM (Tree) [31]	73.4	2017
ST-LSTM(Tree)+Trust Gate [31]	76.5	2017
VA-LSTM [64]	77.5	2017
DPRL [65]	76.9	2018
Bayesian GC-LSTM [43]	82.0	2019
SGP (4L) [46]	78.3	2020
SGP+JCA (4L) [46]	79.2	2020
SR-TSL [5]	80.7	2020
Part-level GCN [66]	83.7	2020
SMAM-Net [17]	75.7	2022
Action-Embed-TR **(Ours)**	**86.4**	2023

are the classes that benefit most from temporal attention. Table 1 shows a comparison for these three settings, where setting-1 largely outperformed others.

4.3.2. Transformer depth

We assessed the depth of Transformers and explained the impact of varying attention layers on network performance. Each attention layer included eight attention heads. Table 2 presents the validation accuracy of the proposed model by using different attention layers 1, 2, 4, and 8. From the comparison, it is observed that the difference in performance is not as significant since NTU-RGB+D videos are relatively short in duration, i.e., around 10 s. From this experiment, we can infer that larger receptive fields of Transformers can benefit in processing longer videos.

4.3.3. Link prediction using different methods

We investigated different link prediction methods for skeleton-based action recognition. First, DeepWalk is selected as the link prediction method for skeleton-based action recognition. DeepWalk predicted new relationships and edges in intra-frame and inter-frames. The links predicted using action embedding correspond to the nodes, which have some action relationships and temporal relationships. Action relationships figured out the interaction between nodes which significantly collaborate for performing some action. In a similar fashion, we evaluated node2vec and GCN techniques for skeleton-based action recognition. Table 3 reports the mean accuracy and computational complexity for these three methods. It demonstrates that the GCN method performs better than others due to its better graph representation capability for skeleton data.

Table 5
Performance Comparison with the state-of-the-art architectures for the NTU-RGB+D dataset.

Method	GFLOPs	Time (ms)	X-Sub	X-View
ST-GCN [42]	8.37	3.426	81.5	88.3
Deep progressive [65]	–	–	82.3	87.7
Actional-GCN [67]	–	–	86.8	94.2
2s Adapt-GCN [9]	35.8	8.862	88.5	95.1
MV-IGNet [68]	–	1.630	89.2	96.3
Shift-GCN [12]	10.0		89.7	96.0
Graph De-Conv [69]	–	–	89.7	95.9
Directed GNN [10]	126.8		89.9	96.1
Decoupling-GCN [70]	16.2	12	90.8	96.6
Info-GCN, [71]	–	–	**93.0**	**97.1**
ST-Transformer [26]	–	–	89.9	96.1
ST-Transformer [27]	–	–	90.3	96.3
Action-Embed-TR **(Ours)**	18.3	7.634	91.5	96.8

Table 6
Performance comparison with the state-of-the-art architectures for the NTU-RGB+D-120 dataset. Best accuracy results are marked in bold.

Method	X-Sub	X-Set
ST-GCN [42]	72.4	71.3
2s AS-GCN [67]	77.7	78.9
MV-IGNet [68]	83.9	85.6
2s Shift-GCN [12]	85.3	86.6
Graph De-Conv [69]	80.8	82.3
Decoupling-GCN [70]	86.5	88.1
Info-GCN, [71]	**89.8**	**91.2**
ST-Transformer [26]	81.9	84.1
ST-Transformer [27]	85.1	87.1
Action-Embed-TR **(Ours)**	**87.7**	**88.5**

4.4. Comparison against the state-of-the-art

We made a performance comparison for currently-available small and large-scale datasets. During our analysis, we found that our method achieves state-of-the-art results.

Comparison over small datasets: For small-scale dataset comparison, we evaluated our method on the SYSU-3D dataset. We used a cross-subject evaluation setting, in which training and testing sets contained random and even distribution of all subjects. SYSU-3D is considered to be a challenging dataset with a large number of subjects, and each action is performed by a single subject for one-time only, thus resulting in large variations. Table 4 shows a comparison and improved performance of our method over other approaches in the literature. This increase in performance demonstrates that the action-embedded self-attention Transformer is effective for skeleton-based action recognition, where action embedding emphasizes spatial features while Transformer architecture preserves temporal features.

Comparison over large datasets: To further assess the generalization of our proposed method, we compare the performance of our proposed method with other state-of-the-art methods for NTU-RGB+D and NTU-RGB+D-120, as shown in Tables 5 and 6, respectively. For a fair comparison, we report the results from original papers in Tables above. Comparing with [26,27], our method is primarily a GCN-based single-stream approach that involves a Transformer network for modeling temporal features. Meanwhile, [26,27] implemented spatial and temporal clues as two separate stream convolution networks using Transformer, enhancing the complexity of their models. We also compare our work with [71] in which information bottleneck learning is investigated for spatial modeling of skeleton using self-attention. We argue that our method is better in comparison with [71], which predominately requires a multi-modal representation of body skeleton. Overall, our proposed model outperforms others for the above-mentioned datasets,

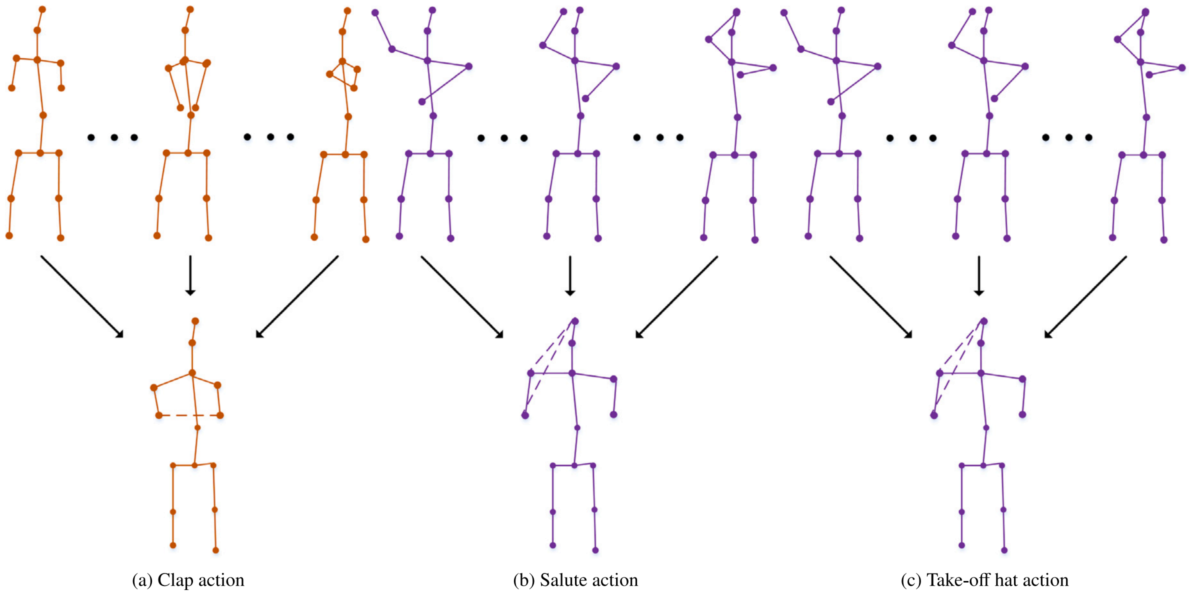


Fig. 6. Visualization of action embedding features for different actions. In clap action, (a) the movement of hands shows the relation of action between two distant joints. Salute and take-off hat actions are similar as the right hand of the skeleton moves close to the head at different speeds.

and thus it establishes the superiority of our model. In Table 5, we also list the computational resources in terms of GFLOPs and inference Time and claim that our method entails reasonable complexity and faster inference time compared to other methods.

5. Visualization

Fig. 6 visualizes the feature maps for different actions in order to validate the effect of these feature maps. The dashed lines demonstrate that the action relationship exists between unconnected joints; therefore, this relationship can be emphasized. For clap-action in Fig. 6(a) from the starting frame, two hands come close and move away, where this articulation is modeled by action embedding and visualized as dotted lines. Likewise, for the salute action in Fig. 6(b), the right hand of the subject move close to the head, encoded as action-relationship by action embedding. The same justification works for the take-off hat action in Fig. 6(c), where the subject's right hand moves closer to the head but relatively at a slower temporal speed.

6. Conclusion

The use of skeleton data for action recognition offers significant potential for developing deep learning-based computer applications. In this work, we propose a novel methodology for skeleton-based action recognition using action-embedding and self-attention Transformer. Action embedding builds the spatial relationship between body joints using link prediction. After that Transformer network models the temporal relationship between joints. The self-attention Transformer receives the input from action embedding block, and the output of Transformer network is fed to MLP for final action classification. The proposed method is evaluated on SYSU-3D, NTU RGB+D and NTU RGB+D120 benchmark datasets, where it outperforms state-of-the-art methods.

Our proposed method only applies to skeleton data, and we utilized the vanilla Transformer architecture in our methodology. In the future, more advanced Transformer architectures with combination of different positional encodings, such as Laplacian positional encoding, will be explored.

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.

Data availability

All datasets are publically available.

References

[1] J.-F. Hu, W.-S. Zheng, J. Lai, J. Zhang, Jointly learning heterogeneous features for RGB-D activity recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5344–5352.

[2] A. Shahroudy, J. Liu, T.-T. Ng, G. Wang, Ntu rgb+ d: A large scale dataset for 3d human activity analysis, in: CVPR, 2016, pp. 1010–1019.

[3] J. Liu, A. Shahroudy, M.L. Perez, G. Wang, L.-Y. Duan, A.K. Chichung, Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding, IEEE Trans. Pattern Anal. Mach. Intell. (2019).

[4] J. Chen, R.D.J. Samuel, P. Poovendran, LSTM with bio inspired algorithm for action recognition in sports videos, Image Vis. Comput. 112 (2021) 104214.

[5] C. Si, Y. Jing, W. Wang, L. Wang, T. Tan, Skeleton-based action recognition with hierarchical spatial reasoning and temporal stack learning network, Pattern Recognit. 107 (2020) 107511.

[6] W. Yang, T. Lyons, H. Ni, C. Schmid, L. Jin, J. Chang, Leveraging the path signature for skeleton-based human action recognition, 2017, arXiv preprint arXiv:1707.03993.

[7] Y. Li, Z. He, X. Ye, Z. He, K. Han, Spatial temporal graph convolutional networks for skeleton-based dynamic hand gesture recognition, EURASIP J. Image Video Process. 2019 (1) (2019) 78.

[8] A. Barkoky, N.M. Charkari, Complex network-based features extraction in RGB-D human action recognition, J. Vis. Commun. Image Represent. 82 (2022) 103371.

[9] L. Shi, Y. Zhang, J. Cheng, H. Lu, Two-stream adaptive graph convolutional networks for skeleton-based action recognition, in: CVPR, 2019, pp. 12026–12035.

[10] L. Shi, Y. Zhang, J. Cheng, H. Lu, Skeleton-based action recognition with directed graph neural networks, in: CVPR, 2019, pp. 7912–7921.

[11] Z. Liu, H. Zhang, Z. Chen, Z. Wang, W. Ouyang, Disentangling and unifying graph convolutions for skeleton-based action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 143–152.

[12] 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.

[13] H. Cai, V.W. Zheng, K.C.-C. Chang, A comprehensive survey of graph embedding: Problems, techniques, and applications, IEEE Trans. Knowl. Data Eng. 30 (9) (2018) 1616–1637.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.

[16] N. Kanwal, T. Eftestøl, F. Khoraminia, T.C.M. Zuiverloon, K. Engan, Vision transformers for small histological datasets learned through knowledge distillation, in: Advances in Knowledge Discovery and Data Mining, Springer Nature Switzerland, 2023, pp. 167–179.

[17] Z. Li, X. Gong, R. Song, P. Duan, J. Liu, W. Zhang, SMAM: Self and mutual adaptive matching for skeleton-based few-shot action recognition, IEEE Trans. Image Process. 32 (2022) 392–402.

[18] U. Asif, D. Mehta, S. Von Cavallar, J. Tang, S. Harrer, DeepActsNet: A deep ensemble framework combining features from face, hands, and body for action recognition, Pattern Recognit. 139 (2023) 109484.

[19] P. Goyal, E. Ferrara, Graph embedding techniques, applications, and performance: A survey, Knowl.-Based Syst. 151 (2018) 78–94.

[20] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710.

[21] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: Large-scale information network embedding, in: Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 1067–1077.

[22] A. Grover, J. Leskovec, Node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855–864.

[23] L.F. Ribeiro, P.H. Saverese, D.R. Figueiredo, Struc2vec: Learning node representations from structural identity, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 385–394.

[24] S. Ivanov, E. Burnaev, Anonymous walk embeddings, 2018, arXiv preprint arXiv:1805.11921.

[25] A. Galland, M. Lelarge, Invariant embedding for graph classification, in: ICML 2019 Workshop on Learning and Reasoning with Graph-Structured Representations, 2019.

[26] C. Plizzari, M. Cannici, M. Matteucci, Spatial temporal transformer network for skeleton-based action recognition, in: Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part III, Springer, 2021, pp. 694–701.

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

[28] R. Girdhar, J. Carreira, C. Doersch, A. Zisserman, Video action transformer network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 244–253.

[29] D. Neimark, O. Bar, M. Zohar, D. Asselmann, Video transformer network, 2021, arXiv preprint arXiv:2102.00719.

[30] J. Zhang, J. Shao, R. Cao, L. Gao, X. Xu, H.T. Shen, Action-centric relation transformer network for video question answering, IEEE Trans. Circuits Syst. Video Technol. (2020).

[31] 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 (4) (2017) 1586–1599.

[32] J. Liu, A. Shahroudy, D. Xu, G. Wang, Spatio-temporal lstm with trust gates for 3d human action recognition, in: European Conference on Computer Vision, Springer, 2016, pp. 816–833.

[33] Q. Ke, S. An, M. Bennamoun, F. Sohel, F. Boussaid, Skeletonnet: Mining deep part features for 3-d action recognition, IEEE Signal Process. Lett. 24 (6) (2017) 731–735.

[34] Q. Ke, M. Bennamoun, S. An, F. Sohel, F. Boussaid, Learning clip representations for skeleton-based 3d action recognition, IEEE Trans. Image Process. 27 (6) (2018) 2842–2855.

[35] C. Hong, J. Yu, J. Zhang, X. Jin, K.-H. Lee, Multimodal face-pose estimation with multitask manifold deep learning, IEEE Trans. Ind. Inform. 15 (7) (2018) 3952–3961.

[36] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE Trans. Image Process. 24 (12) (2015) 5659–5670.

[37] Y. Ou, Z. Chen, 3D deformable convolution temporal reasoning network for action recognition, J. Vis. Commun. Image Represent. 93 (2023) 103804.

[38] J. Yu, M. Tan, H. Zhang, Y. Rui, D. Tao, Hierarchical deep click feature prediction for fine-grained image recognition, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2) (2019) 563–578.

[39] X. Shu, J. Tang, G. Qi, W. Liu, J. Yang, Hierarchical long short-term concurrent memory for human interaction recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2019).

[40] X. Shu, L. Zhang, G.-J. Qi, W. Liu, J. Tang, Spatiotemporal co-attention recurrent neural networks for human-skeleton motion prediction, IEEE Trans. Pattern Anal. Mach. Intell. (2021).

[41] Y. Chen, B. Guo, Y. Shen, W. Wang, W. Lu, X. Suo, Boundary graph convolutional network for temporal action detection, Image Vis. Comput. 109 (2021) 104144.

[42] S. Yan, Y. Xiong, D. Lin, Spatial temporal graph convolutional networks for skeleton-based action recognition, in: AAAI, 2018.

[43] R. Zhao, K. Wang, H. Su, Q. Ji, Bayesian graph convolution lstm for skeleton based action recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6882–6892.

[44] T. Ahmad, L. Jin, L. Lin, G. Tang, Skeleton-based action recognition using sparse spatio-temporal GCN with edge effective resistance, Neurocomputing 423 (2021) 389–398.

[45] W. Peng, X. Hong, H. Chen, G. Zhao, Learning graph convolutional network for skeleton-based human action recognition by neural searching, in: AAAI, 2020, pp. 2669–2676.

[46] Y. Chen, G. Ma, C. Yuan, B. Li, H. Zhang, F. Wang, W. Hu, Graph convolutional network with structure pooling and joint-wise channel attention for action recognition, Pattern Recognit. (2020) 107321.

[47] K. Liu, L. Gao, N.M. Khan, L. Qi, L. Guan, A multi-stream graph convolutional networks-hidden conditional random field model for skeleton-based action recognition, IEEE Trans. Multimed. (2020).

[48] 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.

[49] S. Cho, M. Maqbool, F. Liu, H. Foroosh, Self-attention network for skeleton-based human action recognition, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 635–644.

[50] S. Song, C. Lan, J. Xing, W. Zeng, J. Liu, An end-to-end spatio-temporal attention model for human action recognition from skeleton data, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, 2017.

[51] M.E. Newman, A measure of betweenness centrality based on random walks, Social Networks 27 (1) (2005) 39–54.

[52] F. Fouss, A. Pirotte, J.-M. Renders, M. Saerens, Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation, IEEE Trans. Knowl. Data Eng. 19 (3) (2007) 355–369.

[53] R. Andersen, F. Chung, K. Lang, Local graph partitioning using pagerank vectors, in: 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06), IEEE, 2006, pp. 475–486.

[54] O. Levy, Y. Goldberg, I. Dagan, Improving distributional similarity with lessons learned from word embeddings, Trans. Assoc. Comput. Linguist. 3 (2015) 211–225.

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

[56] N. Kanwal, G. Rizzo, Attention-based clinical note summarization, in: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing, 2022, pp. 813–820.

[57] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D.F. Wong, L.S. Chao, Learning deep transformer models for machine translation, 2019, arXiv preprint arXiv:1906.01787.

[58] A. Baevski, M. Auli, Adaptive input representations for neural language modeling, 2018, arXiv preprint arXiv:1809.10853.

[59] D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus), 2016, arXiv preprint arXiv:1606.08415.

[60] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, D. Tran, Image transformer, in: International Conference on Machine Learning, PMLR, 2018, pp. 4055–4064.

[61] I. Bello, B. Zoph, A. Vaswani, J. Shlens, Q.V. Le, Attention augmented convolutional networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3286–3295.

[62] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European Conference on Computer Vision, Springer, 2020, pp. 213–229.

[63] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst. 32 (2019) 8026–8037.

[64] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, N. Zheng, View adaptive recurrent neural networks for high performance human action recognition from skeleton data, in: CVPR, 2017, pp. 2117–2126.

[65] Y. Tang, Y. Tian, J. Lu, P. Li, J. Zhou, Deep progressive reinforcement learning for skeleton-based action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5323–5332.

[66] L. Huang, Y. Huang, W. Ouyang, L. Wang, Part-level graph convolutional network for skeleton-based action recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 11045–11052.

[67] 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 Conference on Computer Vision and Pattern Recognition, 2019, pp. 3595–3603.

[68] M. Wang, B. Ni, X. Yang, Learning multi-view interactional skeleton graph for action recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2020).

[69] W. Peng, J. Shi, G. Zhao, Spatial temporal graph deconvolutional network for skeleton-based human action recognition, IEEE Signal Process. Lett. 28 (2021) 244–248.

[70] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, H. Lu, Decoupling gcn with dropgraph module for skeleton-based action recognition, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16, Springer, 2020, pp. 536–553.

[71] 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.

Tasweer Ahmad received the bachelor's degree in electrical engineering from the University of Engineering and Technology, Taxila, Pakistan, in 2007, and the master's degree from the University of Engineering and Technology, Lahore, Pakistan, in 2009. He studied for the Ph.D. degree at South China University of Technology, China. He has been an Instructor at the Government College University, Lahore, from 2010 to 2015, and with the COMSATS University Islamabad, Sahiwal Campus, Pakistan, from 2015 to 2016. His current research interests include image processing, computer vision, and machine learning.

Syed Tahir Hussain Rizvi received the Ph.D. degree in computer and control engineering from Politecnico di Torino, Italy, in 2018. He is an experienced researcher and instructor with a demonstrated history of working in the academia and industry. From 2021 to 2023, he worked as a Post-Doctoral Researcher in Department of Electronics and Telecommunications of Politecnico di Torino, Italy on a funded industrial project by Telecom Italia. He recently joined the University of Stavanger as Researcher in Image Processing and Machine Learning. His research interests include the efficient realization of algorithms on embedded systems and applying machine learning to real world problems.

Neel Kanwal received bachelor's degree in electronics engineering from the National University of Computer and Emerging Sciences Pakistan in 2015 and M.Sc. degree in communication and computer networks engineering from the Politecnico di Torino, Italy, in 2020. He is currently pursuing a Ph.D. degree at the University of Stavanger (UiS), Norway. He worked as a Laboratory Engineer at FAST University, Karachi. He is also a Biomedical Data Analysis Laboratory member at the Department of Electrical Engineering and Computer Science, UiS. His research interests include preprocessing, segmentation, and anonymization of histological whole slide images.