Spatio-temporal hard attention learning for skeleton-based activity recognition

Bahareh Nikpour [a,b,*], Narges Armanfard [a,b]

[a] Department of Electrical and Computer Engineering, McGill University Canada
[b] Mila-Quebec AI Institute, Montreal, Quebec, Canada

ARTICLE INFO

Article history:
Received 14 July 2021
Revised 28 January 2023
Accepted 12 February 2023
Available online 18 February 2023

Keywords:
Temporal attention
Spatial attention
Spatio-temporal attention
Activity recognition
Skeleton data
Deep reinforcement learning

ABSTRACT

The use of skeleton data for activity recognition has become prevalent due to its advantages over RGB data. A skeleton video includes frames showing two- or three-dimensional coordinates of human body joints. For recognizing an activity, not all the video frames are informative, and only a few key frames can well represent an activity. Moreover, not all joints participate in every activity; i.e., the key joints may vary across frames and activities. In this paper, we propose a novel framework for finding temporal and spatial attentions in a cooperative manner for activity recognition. The proposed method, which is called STH-DRL, consists of a temporal agent and a spatial agent. The temporal agent is responsible for finding the key frames, i.e., temporal hard attention finding, and the spatial agent attempts to find the key joints, i.e., spatial hard attention finding. We formulate the search problems as Markov decision processes and train both agents through interacting with each other using deep reinforcement learning. Experimental results on three widely used activity recognition benchmark datasets demonstrate the effectiveness of our proposed method.

1. Introduction

Human activity recognition is a popular challenging research direction in the field of computer vision due to its wide range of real-world applications such as human-robot interaction, video understanding, sports analysis, and activity monitoring in older adults [1,2]. Primarily, activity recognition methods were designed based on RGB data; however, with the development of depth cameras, such as Microsoft Kinect and various well-performing human pose estimation methods like [3,4], and [5], recording and extracting skeleton data has become more convenient [1]. Pose estimation is an important problem where the goal is to determine the positions and orientations of different body parts in video frames, which generates the skeleton data [6]. Skeleton data, which contains two- or three-dimensional coordinates of the key body joints (e.g., hand, foot, and neck), has several advantages over RGB data, including but not limited to being robust against viewpoints variations, background noise, and clutter [7]. As such, skeleton-based activity recognition has gained much attention recently. Moreover, the complementary attributes of these two modalities, i.e., RGB and skeleton data, have sparked researchers to take advantage of both when designing their models [8].

Most of the research on skeleton-based activity recognition focuses on feature designing to capture spatial and temporal dynamics of video [9]. However, they assume all the body joints and video frames are equally important while activity could be recognized using only a few key frames [7,10,11]. Hence discarding the redundant frames reduces computational complexity and may improve recognition performance. In addition, not all joints participate in each activity. For example, in the activity “clap”, the upper body joints, such as the hand and wrist, are mainly involved in defining and discriminating the activity.

In this paper, we hypothesize that excluding irrelevant frames and irrelevant body joints improves recognition performance. Motivated by this, we propose a novel framework to simultaneously identify and select relevant frames and joints within a given video. We refer to the joint and frame selection processes as spatial and temporal hard attention findings, respectively. Different from soft attention finding methods which try to assign weights to different parts of data to illustrate their importance, hard attention finding methods aim at removing the irrelevant parts and keeping the important ones, i.e., assigning one and zero weights to them. The proposed framework consists of two agents: a temporal agent for finding the temporal hard attention (i.e., frame selection) and a spatial agent for finding the spatial hard attention (i.e., joint selection).

[*] Corresponding author.
E-mail addresses: bahareh.nikpour@mail.mcgill.ca (B. Nikpour), narges.armanfard@mcgill.ca (N. Armanfard).

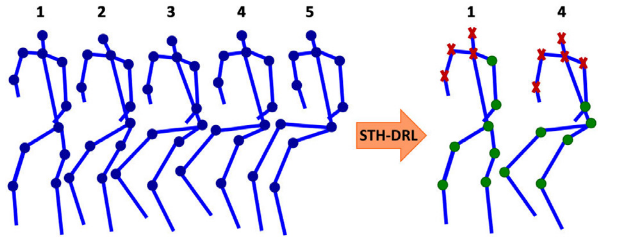


Fig. 1. Motivation of the proposed STH-DRL method. The number above each frame shows its index in the video. The two frames 1 and 4 (out of the five available ones) with a subset of joints, shown by green circles, can well represent the activity “sitting”. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The two agents are trained cooperatively by interacting with each other. First, the temporal agent gets a sequence of frames as input and outputs an indicator vector showing the relevant frames. The relevant frames are selected and fed into the spatial agent that outputs an indicator matrix showing the relevant joints per frame. Figure 1 shows the motivation of the proposed method on activity “sitting” where lower body joints of a couple of frames can well represent the activity. Finally, the selected frames with the selected joints are fed to a baseline activity classifier that recognizes the activity happening in the given video sequence. As the hard attention model is non-differentiable [10], we cannot train the agents in an end-to-end manner. Thus, we formulate the hard-attention-finding problems with Markov Decision Process (MDP) and train each agent using Deep Reinforcement Learning (DRL). Throughout this paper, we refer to the proposed framework for the Spatio-Temporal Hard attention finding using Deep Reinforcement Learning as STH-DRL.

The proposed STH-DRL method is the first study that simultaneously finds spatial and temporal hard attention. STH-DRL does not need any extra ground-truth labels denoting relevant joints and frames; i.e., similar to the traditional recognition models, the only required labels are the video-level activity labels. The training process of the STH-DRL method is supervised by rewards that the agents receive from the baseline classifier. STH-DRL can be considered as a pre-processing technique that can improve the classification performance of the baseline classifier by keeping only the relevant frames/joints and discarding the irrelevant ones. Hence, the proposed framework is capable of improving the recognition performance of the existing activity recognition models if they are employed as the baseline classifier. Such capability can also speed up the training process of the baseline model since the irrelevant information does not contribute to the training. There are some recently proposed methods for finding attention in videos, such as [12] and [13], which are designed for group activity recognition in RGB videos, and [14] where spatio-temporal attention is found to predict motion in RGB videos. However, to the best of our knowledge, this is the first study that proposes the problem of finding spatio-temporal hard attention for skeleton-based activity recognition and solves it by deep reinforcement learning.

The rest of the paper is structured as follows: In Section 2, some related works to our method are reviewed. In Section 3, the proposed STH-DRL method is explained in detail. Section 4 presents the experimental results, and the conclusion is finally drawn in Section 5.

2. Related works

2.1. Skeleton-based activity recognition methods

Human activity can be successfully recognized from skeleton joints’ trajectories; therefore, much research has been accomplished in this area. Finding discriminative features play a key role in activity recognition performance. Earlier methods focus on designing hand-crafted features; e.g., [15] models the three-dimensional relationship between body parts by translations and rotations, in [16] two different kernels are used for the tensor representation of 3D body joints, and [17] employs covariance matrices of joint trajectories. The great capability of deep learning methods in finding effective representations for different applications, such as object recognition [18], led researchers to use it for human activity recognition [19]. Deep learning-based methods for skeleton-based activity recognition use either recurrent neural networks (RNN), convolutional neural networks (CNN), or graph-based networks [9].

RNN is highly powerful for modeling sequences, making it an appropriate choice for video sequence analysis [20]. An RNN-based model with two-stream architecture is proposed in [21], which models temporal and spatial information. Du et al. proposed a hierarchical RNN-based method where the body joints are divided into five subsets regarding the body’s physical structure [22]. Then each subset is given to an individual sub-net as input. The output of the subnets is then hierarchically fused in further layers to reach a final representation. In [23], the human body is divided into individual parts, and the memory cell of a long short-term memory (LSTM) model is split into sub-cells according to the body parts to learn patterns for each of them separately. The final output is then derived out of the combination of the sub-cells. To learn the co-occurrence of skeleton joints, [24] suggests a deep LSTM network along with a regularization method. Lee et al. first transformed the skeleton joints into a new coordinate system to make the representation robust to rotation, translation, and scale [25]. Then the data is fed to an LSTM with short-term, medium-term, and long-term components. The outputs of these components are averaged at the end to derive temporal features. Song et al. designed spatial and temporal attention modules for a multi-layered LSTM network and trained them jointly [7]. In [26], spatial and temporal attention is designed for Lie groups for skeleton-base activity recognition. LSTM is then employed for learning important temporal information about the video sequence.

Two- or three-dimensional coordinates of skeleton joints can be treated as pseudo-images so that CNN-based models can be used to analyze video data. In [27], the position and velocity information of joints are incorporated and fed to a two-stream CNN architecture without considering the long-term dependency of frames. To include temporal information, Ke et al. presented an approach that creates clips out of videos and then gives them to a CNN-based network as input [28]. Liu et al. proposed a view-invariant representation and an enhanced visualization of skeleton data to be employed as the CNN’s input [29]. In [30], a CNN-based recognition method for multi-subject activities is introduced, which uses a hierarchical framework to find co-occurrence features. Banerjee et al. designed four complementary representations for skeleton data as the input of four CNNs. Then, they used a fuzzy approach to fuse the CNNs’ outputs to reach the final decision [31]. Li et. al proposed a novel representation of skeleton video using geometric algebra and then employed CNN to extract features from the new representation of data [32].

The human body can be modeled as a graph considering joints and bones as vertices and edges, respectively. Therefore, employing the graph-based models is beneficial for video analysis. Recently, graph-based methods for activity recognition have become a developing trend in the field of activity recognition [11]. In [33], several spatial-temporal graph convolutions are designed to extract effective features from the skeleton data. Si et al. used a graph convolutional LSTM enhanced with attention to find spatial and temporal information along with their co-occurrence [34]. In [35], a directed graph neural network (DGNN) is proposed, which extracts features from bones and joints as well as their relationship. Also, the paper proposes to change the topological structure of the graph adaptively in the process of training. Liu et al. proposed to remove redundant dependencies between neighboring nodes in the skeleton graph and designed a novel graph convolution to directly model the spatial/temporal dependencies [36]. In [37], a decoupling graph convolutional network (DCGCN) is introduced to boost the recognition performance without adding extra cost. Also, to avoid overfitting, a graph-specific regularization technique is presented. Peng et al. proposed a method for graph pooling called Tripool, which aims at maintaining the diversity in the graph and reducing the redundancy of the node [38]. Their pooling technique can be added to any graph-based activity recognition method to improve performance and decrease computational cost. In [39], the dependencies of joints are modeled by attention blocks without knowing the skeleton graph structure. Another approach is introduced in [40], called Spatial-Temporal Transformer network (ST-TR), where the Transformer self-attention operator is employed to model the joints' dependencies. The architecture has two streams, including spatial and temporal. In the spatial stream, spatial information is extracted, and a convolutional network is used for the time dimension, while in the temporal stream, the temporal information is extracted, and a graph convolution is employed to extract spatial information.

The objective of all the aforementioned skeleton-based activity recognition methods is to learn discriminative spatial/temporal features through designing new network architectures. In contrast, in this work, we propose a novel framework that can be used as a filtering block prior to the existing skeleton-based activity recognition methods, such as the DGNN algorithm. Our proposed framework filters out the irrelevant joints and frames prior to the recognition in the testing phase. In [41], a method is proposed for adapting joints number, with the main goal of having an efficient activity recognition method. However, different from our method, the skeleton is transformed into a skeleton with fewer joints with a transformation matrix, which tends to group adjacent joints. That is while we aim at selecting the relevant joints to each activity and discard the rest. Also, they did not consider finding the relevant frames, which also can increase efficiency.

2.2. Reinforcement learning in activity recognition

A reinforcement learning (RL) algorithm enables an agent to learn a desired task or achieve a complex objective by interacting with its environment and getting feedback from it in the form of reward or punishment [42]. Every RL algorithm is associated with an agent exploring the environment. Usually, the environment is modeled as a Markov Decision Process (MDP), and the agent gets a reward from it with respect to its final goal(s), where the objective is to maximize an expected reward. By incorporating reinforcement learning with deep learning, a new category of machine learning techniques has evolved, called deep RL (DRL), to deal with high dimensional state/action spaces [43].

Deep RL has been employed to solve several problems in the field of computer vision, such as video captioning, person identification, visual tracking, face recognition, and action detection [1]. However, there are few research using deep RL for skeleton-based activity recognition. In [44], Chen et al. used a deep RL framework to extract features from different body parts and activate only the features corresponding to activity-related parts. In [45], an RL-based video summarizing technique is proposed that aims at selecting the key frames in long untrimmed RGB videos. In [46], the key frames in RGB videos are found using a multi-agent reinforcement learning framework. In this method, each agent has the duty of seeking one key frame. Another RL-based method for finding the most relevant frames in RGB videos is proposed in [10], which employs an LSTM agent. To the best of our knowledge, DPRL [11] is the only previous study that uses DRL for skeleton-based activity recognition. The DPRL method improves the recognition performance by selecting key frames (i.e., finding hard temporal attention) in skeleton videos, employing graph representation of data, and a graph-based CNN for generating the required reward.

3. Proposed method

The proposed STH-DRL method consists of a temporal agent, which seeks informative frames within a video, and a spatial agent, which selects the dominant joints within each video frame. We model the process of looking for the informative frames/joints as a Markov decision process and solve it with the popular reinforcement learning algorithm Monte Carlo policy gradient, REINFORCE [47]. The block diagram of STH-DRL is depicted in Fig. 2. Each of the agents is in its current state of the environment. Then it takes an action by interacting with the environment and receiving a reward or punishment out of it; this results in a change in the agent's state. The agents learn to reach their desired goal by maximizing the expected reward. Both agents are run for K episodes. In the following, the details of each agent are first explained. Then, the overall framework for jointly training the agents to find spatio-temporal attention is presented.

3.1. Temporal hard attention exploration

The temporal agent aims at finding the discriminating frames, i.e., temporal hard attention, by maximizing the expected reward. In the kth episode, the temporal agent takes action $\mathbf{a}_k$ according to its current state $\mathcal{S}_k$, and the resultant reward R_k. In the following, agent, state, action, and reward of the proposed frame selection process are described; then, the training process with REINFORCE algorithm is explained.

Temporal agent: Any neural network structure compatible with video data can be employed as the temporal agent. Bi-directional LSTM (BiLSTM) has proved to be effective in processing sequential data. In a video, frames are located in a time sequence, so in this study, we use a BiLSTM-based network topped with a fully connected (FC) layer as the temporal agent. In each episode, the state $\mathcal{S}_k$ is fed to the BiLSTM as its input, and then its output is given to the FC layer. The final output is probability vector $\mathbf{p} = \{p^t\}_{t=1}^{T}$ which later defines the action. T is the total number of frames available in the given video.

State: According to the previous studies [34], considering the motion of the body joints along with the body joints' coordinates can improve human activity recognition performance. Hence, we define the state of the temporal agent in the kth episode as $\mathcal{S}_k = \{S^t\}_{t=1}^{T}$ where $S^t = [S_c^t, S_m^t]$. S_c^t denotes the 3D coordinates of the joints in the tth frame and S_m^t is the motion matrix of the tth frame joints, i.e. $S_m^t = S_c^t - S_c^{t-1}$. Adding such motion information can help the temporal agent to figure out which frame has more information when compared with its previous neighbor frame.

Action: The action that the temporal agent is responsible for is to select informative frames. Two types of actions are defined for this agent as 'keep' or 'remove', which are specified based on the output of the FC layer of the agent. The FC layer's output is a vector of probabilities $\mathbf{p} = \{p^t\}_{t=1}^{T}$, indicating the probability of taking action 'keep'. Consider the action vector in the kth episode as $\mathbf{a}_k = \{a_k^t\}_{t=1}^{T}$. If $a_k^t = 0$, the action is 'remove' and the tth frame should be discarded, while a_k^t being 1 means the action is 'keep' and the tth frame should be kept. Elements of the action vector are sampled from a Bernoulli distribution as bellow:

$$\mathbf{a}_k = \{a_k^t \sim \text{Bernoulli}(p^t)\}_{t=1}^{T} \qquad (1)$$

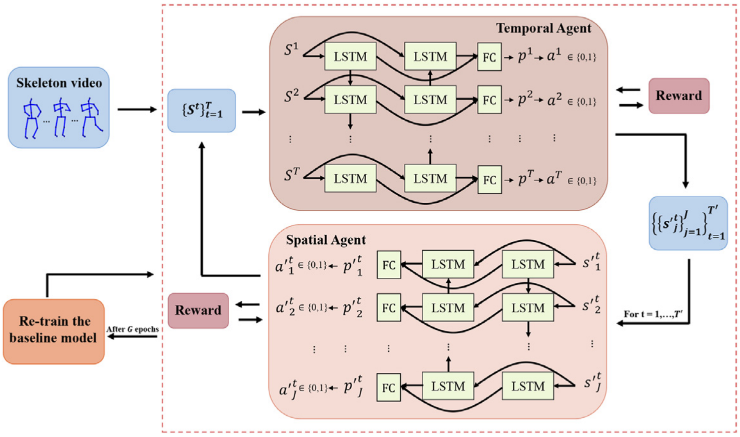

Skeleton video
$\{S^t\}_{t=1}^T$
Temporal Agent
S^1
S^2
S^T
LSTM
FC
$p^1 \rightarrow a^1 \in \{0,1\}$
$p^2 \rightarrow a^2 \in \{0,1\}$
$p^T \rightarrow a^T \in \{0,1\}$
Reward
$\{\{s'^t_j\}_{j=1}^J\}_{t=1}^{T'}$
For t = 1,…,T′
Spatial Agent
$a'^t_1 \in \{0,1\} \leftarrow p'^t_1$
$a'^t_2 \in \{0,1\} \leftarrow p'^t_2$
$a'^t_J \in \{0,1\} \leftarrow p'^t_J$
s'^t_1
s'^t_2
s'^t_J
Reward
After G epochs
Re-train the baseline model

Fig. 2. Overall architecture of the proposed STH-DRL method. S^t is state of the tth frame and $\mathbf{s}'^t_j$ is state of the jth joint in frame t of the input video.

Reward: The reward should guide the temporal agent toward reaching its objective, which is finding the temporal attention; therefore, it should reflect how good the taken action is. To this end, we use a pre-trained baseline recognition model. We feed the selected frames (defined by the action) as input to the baseline pre-trained classifier and calculate the reward based on the baseline prediction. If the class label prediction turns from the wrong label to a correct one, a strong reward Ω is enforced, and if the turning goes from the correct label to a wrong label, a strong punishment $-\Omega$ is enforced. If the predicted class label remains the same, the reward r_0 is determined using the confidence of the baseline model towards predicting the correct class as below:

$$r_0 = \mathrm{sgn}(P_l^k - P_l^{k-1}), \tag{2}$$

where P_l^k is the probability of classifying the input video as class l in the kth episode, and l is the correct class label. Summing up, the Reward at the kth episode, i.e. R_k, is as below:

$$R_k = \begin{cases} \Omega & \text{if reward,} \\ -\Omega & \text{if punishment,} \\ r_0 & \text{otherwise.} \end{cases} \tag{3}$$

Training the temporal agent with REINFORCE: The temporal agent's goal is learning a policy function, which distinguishes informative frames, by maximizing the expected reward $\mathcal{R}(\theta)$ defined as:

$$\mathcal{R}(\theta) = \mathrm{E}_{p_\theta(a_k^{1:T})}[R_k], \tag{4}$$

where $p_\theta(a_k^{1:T})$ is the probability distribution of the possible actions. The policy function is parameterized by θ. According to REINFORCE, the gradient of the expected reward in the kth episode with respect to θ is:

$$\nabla_\theta \mathcal{R}(\theta) = \mathrm{E}_{p_\theta(a_k^{1:T})}[R_k \textstyle\sum_{t=1}^T \nabla_\theta \, ln \, \pi_\theta(a_k^t|S_k^t)], \tag{5}$$

where π_θ denotes the policy function and S_k^t is S^t in the kth episode. We run the temporal agent for K episodes for each skeleton video. Therefore, we can approximate the above gradient by taking the average over gradients of all episodes as below:

$$\nabla_\theta \mathcal{R}(\theta) \approx \tfrac{1}{KT} \textstyle\sum_{k=1}^K [R_k \sum_{t=1}^T \nabla_\theta \, ln \, \pi_\theta(a_k^t|S_k^t)]. \tag{6}$$

To enhance the algorithm's convergence, we reduce the variance by subtracting the average reward of the temporal agent episodes, called b. Hence, the gradient of the reward will be:

$$\nabla_\theta \mathcal{R}(\theta) \approx \tfrac{1}{KT} \textstyle\sum_{k=1}^K [(R_k - b) \sum_{t=1}^T \nabla_\theta \, ln \, \pi_\theta(a_k^t|S_k^t)]. \tag{7}$$

We consider another term in the temporal agent's objective function, alongside maximizing the expected reward $\mathcal{R}(\theta)$. We would like to set an upper bound of M for the number of selected frames. The purpose of adding this term is not to select more than M frames, where M is an integer number between 1 and T and is set by the user. To do so, we consider $\mathbf{1}^T\mathbf{p} \leq M$ as a constraint for the temporal agent's objective function; to be able to solve the optimization problem using gradient descent, the constraint is included in the objective function itself as bellow:

$$\min_\theta -\mathcal{R}(\theta) + \alpha \times (\mathbf{1}^T\mathbf{p} - M), \tag{8}$$

where α is a hyperparameter to control the contribution of its corresponding term.

3.2. Spatial hard attention exploration

The spatial agent is responsible for finding discriminative joints in each frame, i.e., spatial hard attention finding. In the kth episode, the spatial agent is in its current state $\mathcal{S}'_k$, takes action $\mathbf{a}'_k$ and receives reward R'_k. Following [48], the agent, state, action, and reward for the joint selection module are as below:

Spatial agent: Similar to the temporal agent, any neural network structure compatible with video data can be employed as the spatial agent. We can consider the body skeleton as an ordered sequence of body joints, e.g., a sequence starting from the head and ending with the foot. In this sequence, the motion of one joint might affect the others. Therefore, similar to the temporal agent, we employ a BiLSTM-based network topped with a fully connected (FC) layer as the spatial agent. In the kth episode, the spatial agent goes over all the frames once. At frame t, the state S'^t_k is given to the BiLSTM network as input, and then its output is fed to the FC layer. The output of the FC layer is the probability vector $\{p'^t_j\}_{j=1}^J$, defining the action later, where J is the number of joints.

State: Similar to the temporal agent, we use both joint coordinates and motion to define the state of the spatial agent. Therefore, the state of the spatial agent at frame t of the kth episode is $S'^t_k = \{\mathbf{s}'^t_j\}_{j=1}^J$ where $\mathbf{s}'^t_j = [\mathbf{s}'^t_{j,c}, \mathbf{s}'^t_{j,m}]$, and $\mathbf{s}'^t_{j,c}$ denotes the 3D coordinate of the jth joint and $\mathbf{s}'^t_{j,m}$ is the jth joint 3D motion vector, i.e., $\mathbf{s}'^t_{j,m} = \mathbf{s}'^t_{j,c} - \mathbf{s}'^{t-1}_{j,c}$. Considering all the given T' frames at the kth episode, the state set is defined as $\mathcal{S}'_k = \{S'^t_k\}_{t=1}^{T'}$.

Action: The spatial agent's objective is selecting the key joints over the video frames, so we define two actions: 'keep' and 'remove'. To specify the action, we use the output of the FC layer, which shows the probability of taking action 'keep'. In other words, at the tth frame in episode k, the probability $\{p'^t_j\}_{j=1}^J$ shows the probability of action $\{a'^t_{k,j}\}_{j=1}^J$ being 'keep'. Assume $A_k = \{\mathbf{a}'^t_k\}_{t=1}^{T'}$ is the action set in the kth episode over all the T' frames, where $\mathbf{a}'^t_k$ is a J-dim binary vector indicating which joints are selected at frame

t. If $a'^{t}_{k,j}$, which is the jth element of $\mathbf{a}'^{t}_{k}$, is 0, it means the jth joint should be removed; otherwise, it is kept. The elements of $\mathbf{a}'^{t}_{k}$ are sampled from Bernoulli distributions as below:

$$\mathbf{a}'^{t}_{k} = \{a'^{t}_{k,j} \sim \text{Bernoulli}(p'^{t}_{j})\}^{J}_{j=1}. \tag{9}$$

Reward: The reward R'_k in the kth episode for the spatial agent is obtained by feeding the T' frames with the selected joints to the pre-trained baseline model. Similar to the reward calculation process for the temporal agent discussed in Section A, a strong reward (punishment) is given if the predicted class label goes from wrong (correct) to correct (wrong). *ro* is given as the reward if no change happens.

Training the spatial agent with REINFORCE: The spatial agent aims at finding a policy function parameterized by γ, which finds the discriminative joints by maximizing the expected reward $\mathcal{R}'(\gamma)$ as is shown bellow:

$$\mathcal{R}'(\gamma) = \mathrm{E}_{p'_{\gamma}(a'^{1:T'}_{k,1:J})}[R'_k], \tag{10}$$

where $p'_{\gamma}(a'^{1:T'}_{k,1:J})$ is the probability distribution of all the possible actions over the T' frames. The gradient of the expected reward in the kth episode with respect to γ is as follows:

$$\nabla_{\gamma}\mathcal{R}'(\gamma) = \mathrm{E}_{p'_{\gamma}(a'^{1:T'}_{k,1:J'})}[R'_k \textstyle\sum_{t=1}^{T'} \sum_{j=1}^{J} \nabla_{\gamma}\, ln\, \pi_{\gamma}(a'^{t}_{k,j}|\mathbf{s}'^{t}_{k,j})], \tag{11}$$

where π_{γ} denotes the policy function and $\mathbf{s}'^{t}_{k,j}$ is $\mathbf{s}'^{t}_{j}$ in the kth episode. We run the spatial agent for K episodes. The gradient is approximated by taking the average over gradients of the K episodes for each skeleton video as below:

$$\nabla_{\gamma}\mathcal{R}^{'}(\gamma) \approx \frac{1}{KT^{'}} \sum_{k=1}^{K} \left[R^{'}_{k} \sum_{t=1}^{T^{'}} \sum_{j=1}^{J} \nabla_{\gamma}\, ln\, \pi_{\gamma}\left(a^{'t}_{k,j} \middle| \mathbf{s}^{'t}_{k,j}\right) \right]; \tag{12}$$

We then subtract the average rewards of the spatial agent episodes, called b', from all the rewards, to reduce the variance:

$$\nabla_{\gamma}\mathcal{R}'(\gamma) \approx \tfrac{1}{KT'} \textstyle\sum_{k=1}^{K}[(R'_k - b') \sum_{t=1}^{T'} \sum_{j=1}^{J} \nabla_{\gamma}\, ln\, \pi_{\gamma}(a'^{t}_{k,j}|\mathbf{s}'^{t}_{k,j})], \tag{13}$$

To control the number of selected joints, we add a term to the spatial agent's objective function which helps the agent to choose no more than N joints in the frames, where N is a user settable integer number between 1 and J. This can be realized by minimizing $\mathbf{1}^{T}\mathbf{p}' - N$ in the optimization process. Hence, the final objective function is as bellow:

$$\min_{\gamma} -\mathcal{R}'(\gamma) + \beta \times (\mathbf{1}^{T}\mathbf{p}' - N), \tag{14}$$

where $\mathbf{p}'$ is the average probability of the actions over T' frames and β is a hyperparameter to control the contribution of its corresponding term.

3.3. Spatio-Temporal attention algorithm

To have a more effective training process, we first partially pre-train the spatial and temporal agents, independent from each other. The pre-trained agents are then trained mutually– i.e., first, the T frames of the skeleton video are given to the temporal agent, one episode is completed, and T' frames are selected; then, the selected frames are fed to the spatial agent, one episode is completed, and the relevant joints are selected. Both agents complete K episodes, and then their policies are updated.

The pseudo-code of the proposed STH-DRL framework is presented in Algorithm 1. In summary, the baseline classifier is first pre-trained using the original training data (with complete sets of frames and joints). Then each agent is pre-trained independently. Afterward, to train the agents mutually, both agents complete K episodes and update their policies. This process is repeated for G epochs. Then to improve the rewards, the baseline classifier is retrained using the new data (with the selected joints and frames). This procedure is repeated for all the epochs.

Algorithm 1 The proposed STH-DRL method.

```
Input: The training video sequences with labels, baseline recognition classifier, epochs, $K$
Output: Trained temporal agent, Trained spatial agent
Pre-train the baseline model.
Pre-train the temporal agent.
Pre-train the spatial agent.
count = 0.
for epochs do
    count += 1
    for videos do
        for $K$ episodes do
            Run the temporal agent.
            Find the temporal agent's action using (1), take the action and update the state.
            Compute reward of the temporal agent using (2) and (3).
            Run the spatial agent on the selected frames provided by the temporal agent.
            Find the the spatial agent's action using (9), take the action and update the state.
            Compute reward of the spatial agent using (2) and (3).
        end for
        Compute the average reward of temporal agent.
        Compute the loss of the temporal agent using (8).
        Update the temporal agent network parameters.
        Compute the average reward of the spatial agent.
        Compute the loss of spatial agent using (14).
        Update the spatial agent network parameters.
        if count = $G$ then
            Retrain the baseline model using the selected frames and joints.
            count = 0
        end if
    end for
end for
```

4. Experiments

To evaluate the performance of our proposed STH-DRL method, we performed experiments on three widely used activity recognition datasets. We selected three baseline classifiers, each belonging to one of the three activity recognition model categories reviewed in Section 2.1, including CNN, BiLSTM, and the recent advanced graph-based model DGNN [35]. The effectiveness of the STH-DRL method is demonstrated using these three baseline models. Also, the performance of STH-DRL (with DGNN as the baseline) is compared with some state-of-the-art skeleton-based activity recognition methods. In the following, we will explain the datasets we used for our experiments in Section 4.1. Section 4.2 discusses the employed hyperparameters and networks' architectures. We present the recognition results of STH-DRL with different baselines in Section 4.3. Section 4.4 compares the STH-DRL method with state-of-the-art skeleton-based activity recognition methods. The learned temporal and spatial hard attentions are visualized in Section 4.5. The convergence analysis of the spatial and temporal agent networks is presented in Section 4.6. Section 4.7 discusses the effect of hyperparameters α in the temporal agent's objective function and β in the spatial agent's objective function. The sensitivity of STH-DRL to hyperparameters M and N is assessed in Section 4.8. In the end, the effect of STH-DRL on the training run time is investigated in Section 4.9.

4.1. Datasets

NTU+RGBD Dataset (NTU): This is currently the largest publicly available activity recognition dataset with 56,880 video sequences of 4 million frames [23]. 40 subjects participated in capturing the video samples by performing 60 different activities. The NTU data has two train/test split settings. The first setting is Cross-Subject (CS), where 40,320 video samples of 20 subjects are used for training, and the other 16,540 are used as tests. The second setting is Cross-View (CV), in which 37,920 video samples captured from cameras 2 and 3 are used in the train set, and the samples captured from the other camera view, i.e., camera 1, are included in the test set. Each subject is represented by 25 joints, and the videos include either one or two subjects.

UT-Kinect Dataset (UT): In this data, there are 200 video sequences belonging to 10 different activity classes. Each activity is performed by 10 subjects two times [49]. The number of skeleton joints per subject is 20, and all video samples have one subject, i.e., non of the activities are interactive. In this work, the Leave-one-out cross-validation protocol is used for evaluation.

SBU Kinect Interaction Dataset (SBU): This data has 230 video sequences with 6614 frames [50]. There are 8 interactive activity classes, which means there are two persons in all the videos. The number of skeleton joints recorded for each person is 15, so in total, there are 30 joints in each frame. We use the 5-fold cross-validation setting presented for this data for our experiments.

4.2. Implementation details

A BiLSTM with 3 layers is used as each of the agent's networks. We use Adam as the optimizer with initial learning rates 5e-3 and 5e-4 for temporal and spatial agents, respectively. The dropout rate is 0.5. We set hyperparameters Ω, K, α, and β to 25, 6, 0.01, and 0.01, respectively. Hyperparameters M and N are respectively set to half of the number of available frames and joints, i.e., $M = \lceil \frac{T}{2} \rceil$, and $N = \lceil \frac{J}{2} \rceil$ where $\lceil . \rceil$ denotes the ceiling function. The number of epochs for partial pre-training of the temporal and spatial networks is 6. Although the number of epochs for fully training both networks mutually can be selected adaptively based on the values of loss functions defined in (8) and (14), it is set to 9 in all datasets for simplicity. In all datasets, following [11], the bi-cubic interpolation is used to derive video samples with an equal number of frames where the first and last frames remain the same as the original video. The number of frames in all video samples of NTU, UT, and SBU is equalized to respectively 100, 120, and 45. The proposed method is implemented in python using the deep learning framework Pytorch.

4.3. Improving the baseline models

To prove that the proposed spatio-temporal attention finding method boosts the performance of the baseline recognition models, we employed three classifiers, including a BiLSTM, a CNN, and the graph-based method DGNN. The CNN model has 2 layers of convolution and one fully connected layer. The optimizer is Adam. The BiLSTM-based recognition model has 3 layers with a hidden layer size of 256 and the Adam optimization method. For DGNN, for a fair comparison, we used the parameter setting suggested in the original paper. The code available on the respective author's website is used. The hyperparameters of our STH-DRL method are also set to their default values presented in Section 4.2. Recognition performance of the baseline models on the benchmark datasets, before and after applying the STH-DRL method, are shown in Table 1. TH-DRL-X, SH-DRL-X, and STH-DRL-X respectively denote the performance of the partially pre-trained temporal agent, the partially pre-trained spatial agent, and the spatio-temporal attention finding method with baseline model X. On each dataset and among X, TH-DRL-X, SH-DRL-X and STH-DRL-X, the best result is shown in bold. As can be seen, the proposed STH-DRL method has improved the baseline performance for all datasets and baselines. The average accuracy over all datasets for each method is shown in the last column. This column shows that, on average, both pre-trained temporal and spatial agents improve the recognition accuracy of baselines, and training them mutually, i.e., the results obtained by the STH-DRL method, results in the best performance.

Table 1
Activity recognition accuracy (in percent) of the three different baseline models with and without STH-DRL.

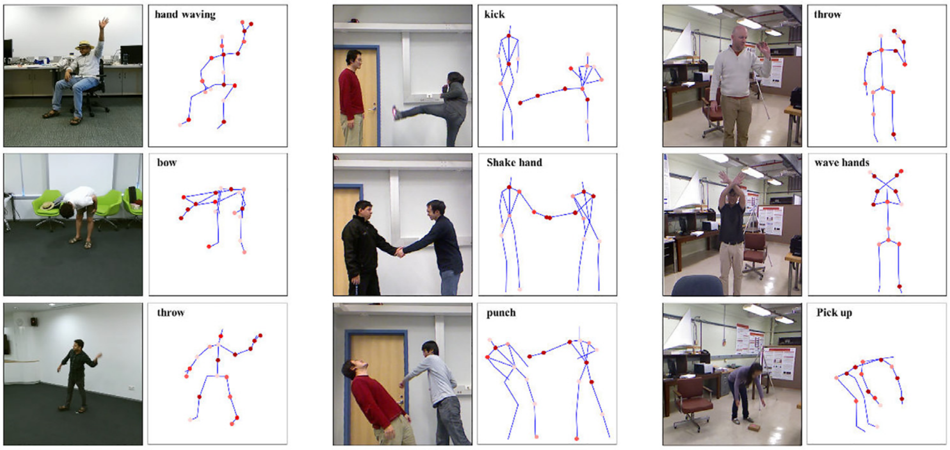
Method	CS	CV	SBU	UT	avg.
BiLSTM	65.0	69.2	76.0	94.5	76.1
TH-DRL-BiLSTM	66.2	68.1	76.8	95.5	76.6
SH-DRL-BiLSTM	67.1	71.3	78.1	95.9	78.1
STH-DRL-BiLSTM	**70.4**	**71.8**	**81.9**	**97.1**	**80.3**
CNN	70.2	71.3	78.2	87.9	76.9
TH-DRL-CNN	68.2	75.9	79.8	92.2	79.0
SH-DRL-CNN	71.8	76.2	85.5	93.1	81.6
STH-DRL-CNN	**75.6**	**81.3**	**86.1**	**95.7**	**84.6**
DGNN	89.9	96.1	87.8	97.5	92.5
TH-DRL-DGNN	87.2	94.8	83.2	95.2	90.1
SH-DRL-DGNN	90.1	96.3	**88.7**	98.4	93.3
STH-DRL-DGNN	**90.8**	**96.7**	**88.7**	**99.1**	**93.8**

Table 2
Activity recognition accuracy (in percent) of different methods on NTU dataset.

Method	CS	CV	year
Lie Group [15]	50.1	52.8	2014
HBRNN [22]	59.1	64.0	2015
Part-aware LSTM [23]	62.9	70.3	2016
LieNet-3Blocks [19]	61.4	67.0	2017
Liu et al. [29]	76	82.56	2017
ST-GCN [33]	81.5	88.3	2018
DPRL+GCNN [11]	83.5	89.8	2018
STA-LSTM [51]	73.4	81.2	2018
DGNN [35]	89.9	96.1	2019
AGC-LSTM [34]	89.2	95.0	2019
DCGCN [37]	88.1	95.2	2020
STA-DeepLG[26]	72.38	79.72	2021
ST-TR [40]	89.9	96.1	2021
3s-AdaSGN [41]	90.5	95.3	2021
STH-DRL	**90.8**	**96.7**	

4.4. Comparison to state-of-the-art

To prove the effectiveness of our proposed method, we adopt DGNN as the baseline model and compare STH-DRL-DGNN with several state-of-the-art skeleton-based methods. Table 2, presents the accuracy comparison of the two settings of the NTU dataset, i.e. CS and CV. As can be seen, our proposed method has the best performance among the eleven other activity recognition algorithms. Performance comparison of the SBU dataset with six state-of-the-art methods is shown in Table 3. This confirms our method yields the best accuracy among all. In Table 4, the results of the STH-DRL method compared to ten other algorithms are shown. As can be seen, the proposed method achieves the best performance for this data, as well.



(a) Three activities of NTU dataset (b) Three activities of SBU dataset (c) Three activities of UT dataset

Fig. 3. Visualizing frequency of selecting a joint by STH-DRL within a video, for three different activity classes from the NTU, SBU and UT datasets. Lighter red color indicates that the corresponding joint is less frequently selected and a darker one indicates higher selection frequency. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Activity recognition accuracy (in percent) of different methods on SBU dataset.

Method	SBU	year
Raw skeleton [52]	49.7	2012
Joint feature [53]	86.9	2014
Hierarchical RNN [22]	80.35	2015
CHARM [54]	83.9	2015
DGNN [35]	87.8	2019
DCGCN [37]	88.2	2020
STH-DRL	**88.7**	

Table 4
Activity recognition accuracy (in percent) of different methods on UT dataset.

Method	UT	year
Histogram of 3D Joints [49]	90.9	2012
Riemannian Manifold [55]	91.5	2015
Grassmann Manifold [56]	88.5	2015
GMSM [57]	97.4	2016
SCK+DCK [16]	98.2	2016
ST-LSTM+Trust Gate [20]	97.0	2017
ST-NBNN [17]	98.0	2017
DPRL+GCNN [11]	98.5	2018
DGNN [35]	97.5	2019
DCGCN [37]	98.2	2020
STA-DeepLG [26]	97.7	2021
STH-DRL	**99.1**	

4.5. Visualization of the learned spatial and temporal hard attention

To visually analyze the results, we show the selected joints for three different activities from the three datasets SBU, UT, and NTU in Fig. 3. The color brightness of the red circles on each joint illustrates the frequency of choosing that joint over the whole given T' frames. The brighter circles indicate the corresponding joints are selected less frequently compared to the joints being specified by a darker color. For example, in activity “kick” in the SBU dataset, the lower body joints are correctly selected in all frames and the irrelevant joints such as the shoulder are correctly removed. The final output of STH-DRL, i.e. the final selected frames along with their corresponding selected joints for activity “hand waving” of NTU data are demonstrated in Fig. 4. The removed frames are shown in gray, and the selected joints are specified with red circles. As can be seen, there are no observable differences between frames 45 to 49; therefore, the algorithm has kept only frame 48 as their representative. Also, between frames 54 and 55, a hand movement is observed, so both frames are kept.

To have a more general view of the selected joints, the distribution of the selected joints, over all the activities and for each of the three datasets, is shown in Fig. 5. This figure indicates that the selected joints by STH-DRL vary among different activities, and not all joints are informative for an activity.

Figures 4, 3, and 5 demonstrate the consistency of the learned spatial and temporal attentions with human perception.

4.6. Convergence analysis

As is discussed in Section 3, temporal and spatial agents aim to minimize the loss functions defined in (8) and (14), respectively. To see the trend of optimization, the loss value of temporal agent and spatial agent versus training iteration, for SBU, UT, and NTU (CS) datasets are shown in Figs. 6 and 7. As these graphs show, at the beginning of the training, the loss oscillates significantly, which is due to the low skill level of agents. However, the oscillation decreases over time as the agents become skilled in choosing important joints and frames. In the end, the losses of both temporal and spatial agents converge to zero, which is desired.

4.7. Effect of hyperparameters α and β

In this section, the influence of hyperparameter α in the loss function of temporal agent shown in (8), and hyperparameter β in the loss function of spatial agent shown in (14) are investigated. Figure 8a shows the accuracy vs. α where α is selected from the set $\{0, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ and β is set to its default value i.e. 10^{-2}. Figure 8b demonstrates the accuracy of STH-DRL vs. β where $\beta \in \{0, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ and α is set to its default value 10^{-2}. STH-DRL with the BiLSTM baseline and the UT dataset is used for this experiment. As can be observed in these graphs, imposing the upper bounds on the number of selected frames and joints is effective in the recognition performance.

4.8. Sensitivity to hyperparameters M and N

The sensitivity of STH-DRL to the hyperparameters M and N are explored in this section. To this end, M is selected

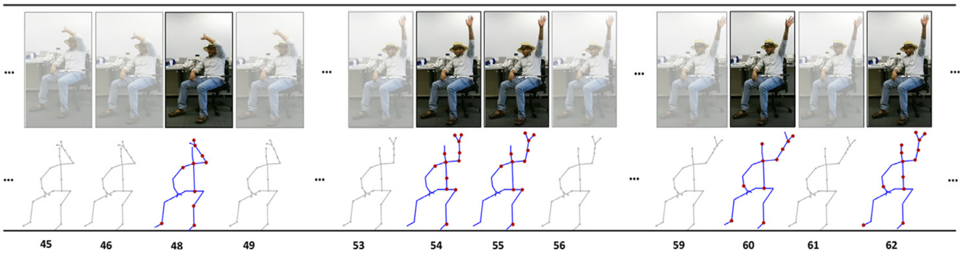
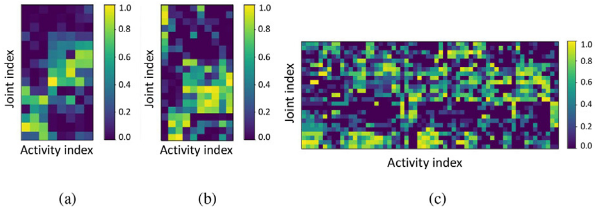


Fig. 4. Visualizing the selected frames and joints by STH-DRL for the activity "hand waving" in the NTU dataset. The frames shown in gray color are discarded from the data. The selected joints are shown in red circles. The index of each frame is shown below the frame. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) (b) (c)

Fig. 5. Distribution of the engaged joints in different actions for the (a) SBU, (b) UT, and (c) NTU(CS) datasets.

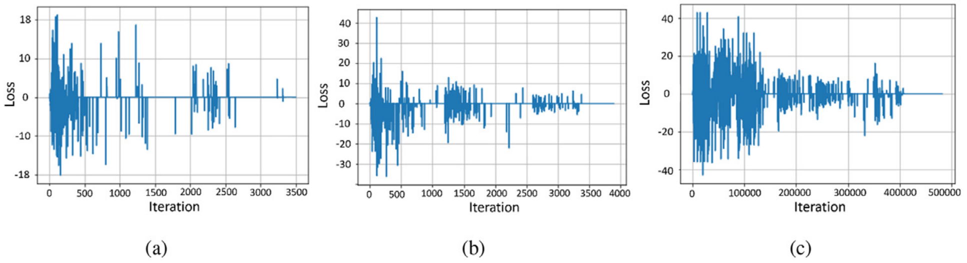
from the sets {3, 8, 16, 22, 30, 38, 45},{5, 15, 30, 60, 85, 105, 120} and {5, 10, 35, 50, 65, 80, 100} for SBU, UT, and NTU datasets, respectively. The accuracy of STH-DRL with BiLSTM baseline vs. M is shown in Fig. 9. As is observed, the best M value for all three datasets is half of the number of available frames. The same experiment is performed for the parameter N where N is selected from the sets {3, 7, 10, 15, 19, 26, 30}, {2, 5, 8, 10, 13, 17, 20} and {5, 10, 15, 25, 30, 40, 50} respectively for the SBU, UT and NTU datasets. Figure 10 shows the recognition accuracy, which indicates that setting N to half of the number of available joints leads to the highest accuracy in all three datasets. Both Figs. 9 and 10 show that the performance of STH-DRL is not too sensitive to the hyperparameters M and N if they are not set to a very low value, which is a desirable property. Moreover, it can be seen that keeping all the available information, i.e., frames and joints, will not always lead to the best accuracy. This is more significant in the NTU dataset, where increasing the upper bound for both frames and joints decreases the recognition accuracy. This can be associated with the fact that the NTU dataset is a more challenging data having many more activity classes compared to the UT and SBU datasets, and there exist several similar classes of activities in the data. Hence, selecting key frames and joints play a more important role in this data. It should be noted that 5-fold and leave-one-out cross-validation procedures are used to compute the accuracy for the SBU and UT datasets, respectively.

4.9. Run time improvement

Table 5
Run time (in hours) of training a BiLSTM-based recognition model with (T_{pr+p}) and without (T_o) STH-DRL.

Method	CS	CV	SBU	UT	avg.
T_{pr}	5.82	5.53	0.03	0.03	2.85
T_p	40.89	43.52	0.19	0.24	21.21
T_{pr+p}	46.71	49.05	0.22	0.27	24.06
T_o	95.70	90.20	0.68	0.61	46.79

Based on our experiments, about 60% of the frames and 55% of the joints are removed in each video by the STH-DRL method. In this section, we investigate the effectiveness of the proposed STH-DRL framework, as a pre-processing block, in reducing the (re-)training time of the baseline classifier. To this end, we first train STH-DRL (with baseline BiLSTM) as is discussed in Section 3. We then apply the trained STH-DRL to the data to identify and select the relevant frames and joints. The elapsed time of this pre-processing phase for each dataset is recorded and shown by T_{pr} in Table 5. Then, the pre-processed data, which includes only the informative frames and joints, is used to train a BiLSTM (initialized randomly) recognition model; the elapsed time for training the BiLSTM is recorded and shown by T_p in the table. In the third row of the table, the total elapsed time, i.e. $T_{pr+p} = T_p + T_{pr}$ is reported. The last row of the table shows the required time T_o for training the same BiLSTM model (with random initial weights) but with the *original* data, i.e., the data with all frames and joints. The same epoch number is used for training BiLSTM in each case. The average of the run times, avg., on all the datasets are shown in the last column of the table. All the run times reported in the table are in hours. Comparing the last two rows of the table indicates our proposed method if used as a pre-processing block, speeds up the training phase of the recognition model on average by about 48%. Faster training time can be associated with two factors: 1) only relevant joints and frames are observed during the training phase of the baseline classifier, and 2) a much less number of network parameters are needed for the pre-processed data, as the result of the frame selection achieved by the temporal agent. Figure 11 shows the histogram of the number of selected frames for video samples of the three datasets SBU, UT, and NTU. As can be seen, the temporal agent removes about 60% of the frames on average in all three data sets, which confirms that our proposed method decreases the run time of the classifier's training procedure by re-



(a) (b) (c)

Fig. 6. Loss of the temporal agent of STH-DRL-BiLSTM vs. iterations for the (a) SBU, (b) UT, and (c) NTU(CS) datasets.

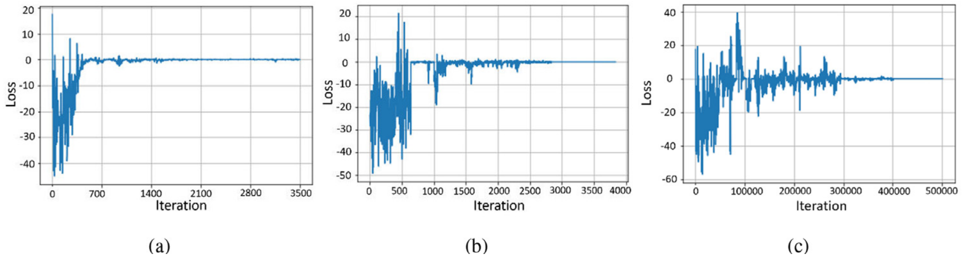


Fig. 7. Loss of the spatial agent of STH-DRL-BiLSTM vs. iterations for the (a) SBU, (b) UT, and (c) NTU(CS) datasets.

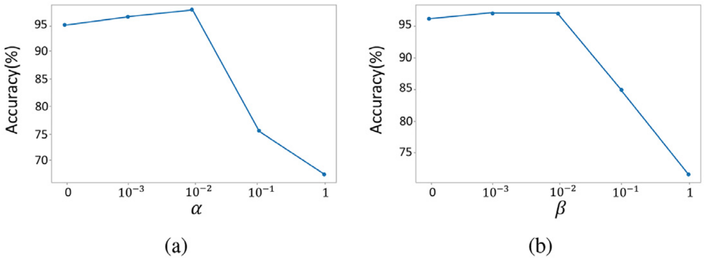


Fig. 8. Accuracy of STH-DRL-BiLSTM vs. (a) α and (b) β for the UT dataset.

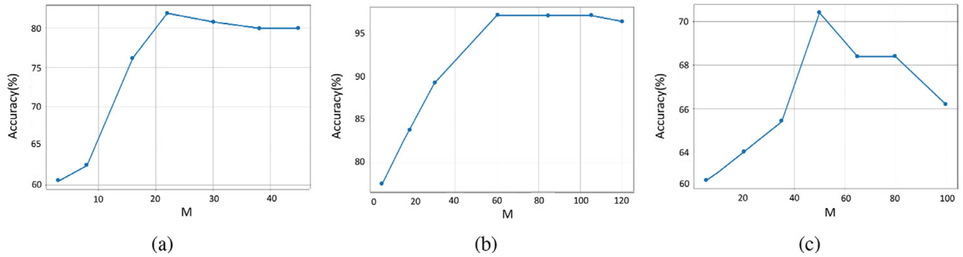


Fig. 9. Accuracy of STH-DRL-BiLSTM vs. M for the (a) SBU, (b) UT, and (c) NTU(CS) datasets.

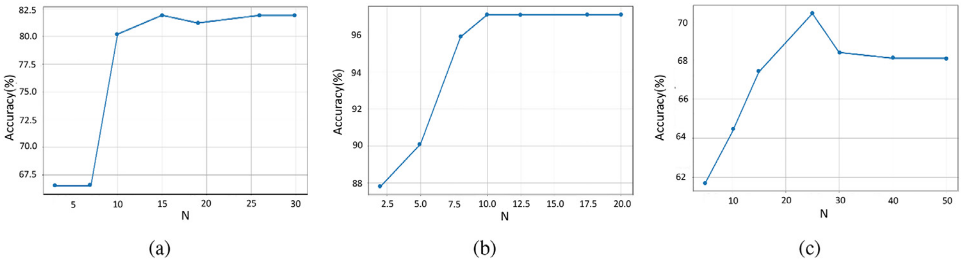


Fig. 10. Accuracy of STH-DRL-BiLSTM vs. N for the (a) SBU, (b) UT, and (c) NTU(CS) datasets.

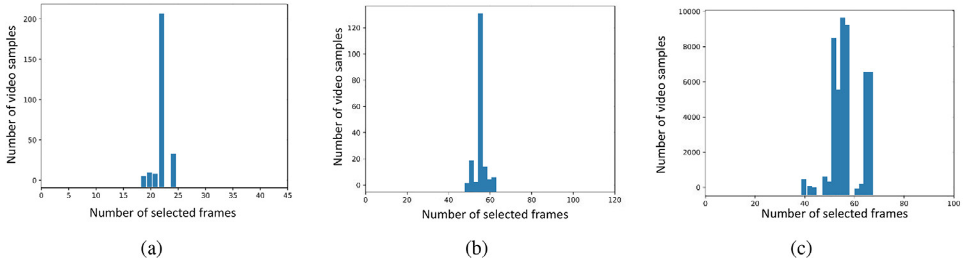


Fig. 11. The histogram of the number of selected frames over the number of video samples for (a) SBU, (b) UT, and (c) NTU datasets.

quiring a smaller number of parameters. Also, the graphs confirm that despite the common belief that including more information (i.e., all frames and joints) results in better performance, considering only relevant frames and joints can still provide the same level of accuracy and also even better performance in several cases. Capability of STH-DRL in reducing the (re-)training run time of an activity classifier is especially beneficial for online activity recognition applications where a fast and effective (re-)training phase is required. One Tesla P100-PCIE-16GB GPU is used to run these experiments.

5. Conclusion

In this work, we proposed a novel spatio-temporal attention finding method, called STH-DRL, which selects the relevant frames and joints in skeleton videos and discards the irrelevant ones, to improve the performance of activity recognition models. In other words, our proposed STH-DRL method finds both spatial and temporal hard attentions in skeleton videos. We formulate the problem as two Markov decision processes and solve them with the popular policy gradient algorithm, REINFORCE. We designed a spatial agent for finding the key joints and a temporal agent to find the key frames. Each agent has its own specified environment, state, and action. The two agents are trained by interacting with each other in order to find their optimal policy. STH-DRL has the capability to be employed prior to the existing human activity recognition models to improve their recognition performance. Three widely used benchmark datasets including NTU, SBU, and UT-Kinect are used in our experiments and performance analyses. We used three recognition methods to demonstrate the effectiveness of the frame and joint selections performed by the STH-DRL method. As our experiments denoted, our method could improve the baseline classifiers' performance by about 4.4% on average. We also compared the proposed method with state-of-the-art skeleton-based activity recognition methods, and the results confirmed the effectiveness of our method. In addition, we demonstrated that STH-DRL, as a pre-processing block, can decrease the training time of a baseline activity classifier. The run time reduction is more specifically beneficial in applications where online training plays an important role. Moreover, frame reduction is useful in video transmission where the transmitter decides which frames to transmit to reduce the transmission time and complexity.

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

Data will be made available on request.

Acknowledgment

The authors wish to acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC). This research was enabled in part by support provided by Calcul Quebec and Compute Canada.

References

[1] B. Nikpour, D. Sinodinos, N. Armanfard, Deep reinforcement learning in human activity recognition: a survey (2022).

[2] M. Naveenkumar, S. Domnic, Deep ensemble network using distance maps and body part features for skeleton based action recognition, Pattern Recognit. 100 (2020) 107125.

[3] W. Li, H. Liu, H. Tang, P. Wang, L. Van Gool, MHFormer: multi-hypothesis transformer for 3D human pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 13147–13156.

[4] W. Li, H. Liu, R. Ding, M. Liu, P. Wang, W. Yang, Exploiting temporal contexts with strided transformer for 3D human pose estimation, IEEE Trans. Multimedia (2022).

[5] D. Pavllo, C. Feichtenhofer, D. Grangier, M. Auli, 3D human pose estimation in video with temporal convolutions and semi-supervised training, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7753–7762.

[6] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, Y. Sheikh, OpenPose: realtime multi--person 2D pose estimation using part affinity fields, IEEE Trans. Pattern Anal. Mach. Intell. 43 (1) (2019) 172–186.

[7] 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.

[8] J. Li, X. Xie, Q. Pan, Y. Cao, Z. Zhao, G. Shi, SGM-Net: skeleton-guided multimodal network for action recognition, Pattern Recognit. 104 (2020) 107356.

[9] B. Ren, M. Liu, R. Ding, H. Liu, A survey on 3D skeleton-based action recognition using learning method, arXiv preprint arXiv:2002.05907(2020).

[10] W. Dong, Z. Zhang, T. Tan, Attention-aware sampling via deep reinforcement learning for action recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 8247–8254.

[11] 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.

[12] X. Shu, L. Zhang, Y. Sun, J. Tang, Host–parasite: graph LSTM-in-LSTM for group activity recognition, IEEE Trans. Neural Netw. Learn. Syst. 32 (2) (2020) 663–674.

[13] X. Shu, L. Zhang, Y. Sun, J. Tang, Host-parasite: graph LSTM-in-LSTM for group activity recognition, IEEE Trans. Neural Netw. Learn. Syst. 32 (2) (2021) 663–674, doi:10.1109/TNNLS.2020.2978942.

[14] 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).

[15] R. Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3D skeletons as points in a lie group, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 588–595.

[16] P. Koniusz, A. Cherian, F. Porikli, Tensor representations via kernel linearization for action recognition from 3D skeletons, in: European Conference on Computer Vision, Springer, 2016, pp. 37–53.

[17] J. Weng, C. Weng, J. Yuan, Spatio-temporal naive-bayes nearest-neighbor (ST-NBNN) for skeleton-based action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4171–4180.

[18] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.

[19] Z. Huang, C. Wan, T. Probst, L. Van Gool, Deep learning on lie groups for skeleton-based action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6099–6108.

[20] J. Liu, A. Shahroudy, D. Xu, A.C. Kot, G. Wang, Skeleton-based action recognition using spatio-temporal LSTM network with trust gates, IEEE Trans. Pattern Anal. Mach. Intell. 40 (12) (2017) 3007–3021.

[21] H. Wang, L. Wang, Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 499–508.

[22] Y. Du, W. Wang, L. Wang, Hierarchical recurrent neural network for skeleton based action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1110–1118.

[23] 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.

[24] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, X. Xie, Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, 2016.

[25] I. Lee, D. Kim, S. Kang, S. Lee, Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1012–1020.

[26] C. Ding, K. Liu, F. Cheng, E. Belyaev, Spatio-temporal attention on manifold space for 3D human action recognition, Appl. Intell. 51 (1) (2021) 560–570.

[27] Y. Du, Y. Fu, L. Wang, Skeleton based action recognition with convolutional neural network, in: 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), IEEE, 2015, pp. 579–583.

[28] Q. Ke, M. Bennamoun, S. An, F. Sohel, F. Boussaid, A new representation of skeleton sequences for 3D action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3288–3297.

[29] M. Liu, H. Liu, C. Chen, Enhanced skeleton visualization for view invariant human action recognition, Pattern Recognit. 68 (2017) 346–362.

[30] C. Li, Q. Zhong, D. Xie, S. Pu, Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation, arXiv preprint arXiv:1804.06055(2018).

[31] A. Banerjee, P.K. Singh, R. Sarkar, Fuzzy integral based CNN classifier fusion for 3D skeleton action recognition, IEEE Trans. Circuits Syst. Video Technol. (2020).

[32] Y. Li, R. Xia, X. Liu, Learning shape and motion representations for view invariant skeleton-based action recognition, Pattern Recognit. 103 (2020) 107293.

[33] 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, Vol. 32, 2018.

[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] L. Shi, Y. Zhang, J. Cheng, H. Lu, Skeleton-based action recognition with directed graph neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7912–7921.

[36] 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.

[37] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, H. Lu, Decoupling GCN with drop-graph module for skeleton-based action recognition, in: European Conference on Computer Vision, Springer, 2020, pp. 536–553.

[38] W. Peng, X. Hong, G. Zhao, Tripool: graph triplet pooling for 3D skeleton-based action recognition, Pattern Recognit. 115 (2021) 107921.

[39] 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.

[40] C. Plizzari, M. Cannici, M. Matteucci, Spatial temporal transformer network for skeleton-based action recognition, in: International Conference on Pattern Recognition, Springer, 2021, pp. 694–701.

[41] L. Shi, Y. Zhang, J. Cheng, H. Lu, AdaSGN: adapting joint number and model size for efficient skeleton-based action recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13413–13422.

[42] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT press, 2018.

[43] V. François-Lavet, P. Henderson, R. Islam, M.G. Bellemare, J. Pineau, An introduction to deep reinforcement learning, arXiv preprint arXiv:1811.12560(2018).

[44] L. Chen, J. Lu, Z. Song, J. Zhou, Part-activated deep reinforcement learning for action prediction, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 421–436.

[45] K. Zhou, Y. Qiao, T. Xiang, Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.

[46] W. Wu, D. He, X. Tan, S. Chen, S. Wen, Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6222–6231.

[47] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Mach. Learn. 8 (3–4) (1992) 229–256.

[48] B. Nikpour, N. Armanfard, Joint selection using deep reinforcement learning for skeleton-based activity recognition, in: 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2021, pp. 1056–1061.

[49] L. Xia, C.-C. Chen, J.K. Aggarwal, View invariant human action recognition using histograms of 3D joints, in: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, IEEE, 2012, pp. 20–27.

[50] K. Yun, J. Honorio, D. Chattopadhyay, T.L. Berg, D. Samaras, Two-person interaction detection using body-pose features and multiple instance learning, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2012.

[51] S. Song, C. Lan, J. Xing, W. Zeng, J. Liu, Spatio-temporal attention-based LSTM networks for 3D action recognition and detection, IEEE Trans. Image Process. 27 (7) (2018) 3459–3471.

[52] K. Yun, J. Honorio, D. Chattopadhyay, T.L. Berg, D. Samaras, Two-person interaction detection using body-pose features and multiple instance learning, in: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, IEEE, 2012, pp. 28–35.

[53] Y. Ji, G. Ye, H. Cheng, Interactive body part contrast mining for human interaction recognition, in: 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), IEEE, 2014, pp. 1–6.

[54] W. Li, L. Wen, M.C. Chuah, S. Lyu, Category-blind human action recognition: a practical recognition system, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4444–4452.

[55] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, A. Del Bimbo, 3D Human action recognition by shape analysis of motion trajectories on Riemannian manifold, IEEE Trans. Cybern. 45 (7) (2014) 1340–1352.

[56] R. Slama, H. Wannous, M. Daoudi, A. Srivastava, Accurate 3D action recognition using learning on the Grassmann manifold, Pattern Recognit. 48 (2) (2015) 556–567.

[57] P. Wang, C. Yuan, W. Hu, B. Li, Y. Zhang, Graph based skeleton motion representation and similarity measurement for action recognition, in: European Conference on Computer Vision, Springer, 2016, pp. 370–385.