Spatial Hard Attention Modeling via Deep Reinforcement Learning for Skeleton-Based Human Activity Recognition

Bahareh Nikpour, *Graduate Student Member, IEEE*, and Narges Armanfard

***Abstract*—Deep learning-based algorithms have been very successful in skeleton-based human activity recognition. Skeleton data contains 2-D or 3-D coordinates of human body joints. The main focus of most of the existing skeleton-based activity recognition methods is on designing new deep architectures to learn discriminative features, where all body joints are considered equally important in recognition. However, the importance of joints varies as an activity proceeds within a video and across different activities. In this work, we hypothesize that selecting relevant joints, prior to recognition, can enhance performance of the existing deep learning-based recognition models. We propose a spatial hard attention finding method that aims to remove the uninformative and/or misleading joints at each frame. We formulate the joint selection problem as a Markov decision process and employ deep reinforcement learning to train the proposed spatial-attention-aware agent. No extra labels are needed for the agent's training. The agent takes a sequence of features extracted from skeleton video as input and outputs a sequence of probabilities for joints. The proposed method can be considered as a general framework that can be integrated with the existing skeleton-based activity recognition methods for performance improvement purposes. We obtain very competitive activity recognition results on three commonly used human activity recognition datasets.**

***Index Terms*—Activity recognition, deep reinforcement learning (DRL), skeleton data, spatial attention.**

I. Introduction

IN THE field of computer vision, activity recognition is a very practical, yet challenging task which plays a significant role in video understanding. Despite decades of study, activity recognition remains extremely popular because of its vast potential applications, e.g., human–robot interaction, monitoring indoor and outdoor activities, video surveillance, and sports analysis [6], [7], [15], [46]. Many attempts have been made to recognize human activity from skeleton data or RGB video images. Skeleton data is a data modality that contains two-dimensional (2-D) or three-dimensional (3-D) coordinates of body joints (i.e., head, neck, . . . , foot). Despite RGB video-based activity recognition methods, which mainly focus on the appearance information, skeleton-based methods are robust against background clutter, illumination changes, appearance variation, etc. It has been shown that an activity can be effectively recognized by tracking the skeleton joints locations across video, without any need to the RGB information [1]. Nowadays skeletal data are easily accessible thanks to the prevalence of depth cameras, such as Microsoft Kinect, and reliable performance of the existing human pose estimation algorithms [38], [48], [54], [57], [62]. Hence, skeleton-based activity recognition has gained increasing attention [17], [41], [52], [55], [58]. This article focuses on human activity recognition using skeleton data.

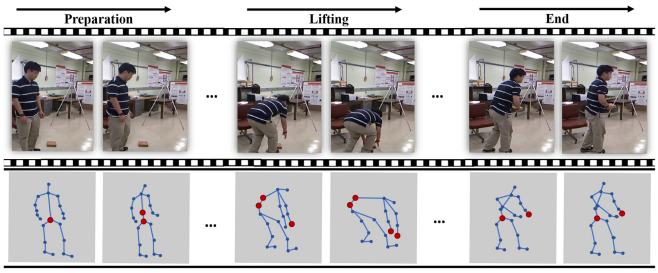

Fig. 1. Different joint subsets (shown in red circles) are involved in different stages of activity "pick up."

Not all human joints are equally important for activity recognition. As examples, consider two activities "throw" and "kick." In activity "throw," the articulated configurations of upper body joints are important, while lower body joints play more role in activity "kick." In addition, the key joints may vary over time as an activity proceeds. The variation of key joints across video frames is illustrated in Fig. 1.

Most of the skeleton-based activity recognition methods assume equal importance for all of the body joints and pay little attention to the fact that, indeed, monitoring of some of the body joints is not essential and even could mislead the recognition process.

We hypothesize that the irrelevant joints across frame/video bring noise and degrade recognition performance; hence, the irrelevant joints should be discarded by hard attention. Motivated by this, we propose a spatial-attention-aware selection agent to continuously identify discriminative joints and

Manuscript received 28 September 2022; accepted 15 February 2023. This article was recommended by Associate Editor Q. Wang. *(Corresponding author: Bahareh Nikpour.)*

The authors are with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 0G4, Canada, and also with 6666 Rue Saint-Urbain, Mila-Quebec AI Institute, Montreal, QC H2S 3H1, Canada (e-mail: bahareh.nikpour@mcgill.ca; narges.armanfard@mcgill.ca).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2023.3250120.

Digital Object Identifier 10.1109/TSMC.2023.3250120

discard the irrelevant ones. The agent takes a sequence of features, extracted from the skeleton video, as input and outputs a sequence of probabilities for joins. Since the hard attention model is nondifferentiable, it cannot be trained in an end-to-end manner [21], [58]. As such, we employ deep reinforcement learning (DRL) to train the spatial-attention-aware agent. Our method has the following benefits: First, the proposed method needs no extra labels denoting irrelevant joints (per frame/video); i.e., only video-level activity labels are required. The agent training process is supervised by a reward generated by a baseline recognition model. Second, the proposed agent is compatible with most of the existing activity recognition models (to be used as the baseline model) and can boost their performance effectively. Third, the proposed agent selects a distinct "optimal"[1] subset of key joints for each video frame. The selected subset of joints may differ both in size and membership across frames. This allows to incorporate in recognition the fact that the important joints may vary across different stages of an activity. Fourth, a trained spatial-attention-aware agent can be used as a preprocessing block, for a recognition model, that filters out irrelevant joints before recognition. Such integration can significantly speed up the training phase of the recognition model since the irrelevant joints are not involved in the recognition model training. The main contributions of our article are listed.

1) We discover the novel problem of finding spatial hard attentions in skeleton video for human activity recognition, using deep learning.
2) To address this problem, we propose a spatial-attention-aware agent that keeps relevant joints and discards irrelevant ones. The agent is trained by DRL.
3) We performed experiments on three widely used benchmark activity recognition datasets to show the effectiveness of our proposed method and achieve competitive results.

Across this article, we refer to the proposed method as SHARL, which stands for Spatial Hard Attention finding using deep RL. To the best of our knowledge, this is the first study that devises a deep learning-based framework for spatial hard attention finding in skeleton video.

An early version of this article is appeared in [71]. The remainder of this article is organized as follows. Section II discusses the related works to our method. Section III introduces the proposed SHARL method. The experiments are presented in Section IV. We draw the conclusion in Section V.

II. Related Works

A. Deep Learning-Based Activity Recognition With Skeleton Data

Extensive research has been carried out on creating appropriate representations for human activity recognition. Great success of deep learning in object recognition [8] motivated researchers to use it in human activity recognition [32] to find effective representations. In the literature, deep learning-based methods for skeleton-based activity recognition are mainly grouped into three categories: methods based on 1) recurrent neural networks (RNNs); 2) convolutional neural networks (CNNs); and 3) graph-based networks [67].

RNN has proved to perform well in video analysis due to it is capability in modeling sequential data [23], [36]. A two-stream RNN-based model is proposed in [42] which captures spatial and temporal information. An end-to-end approach using hierarchical RNN is suggested in [18] where the body skeleton is first divided into five parts. Then, it feeds each of the parts to an individual subnet as input. In [27], a part-aware long short-term memory (LSTM)-based method is presented that considers each body part separately. Zhu et al. [30] employed a deep LSTM network along with a regularization technique for learning the co-occurrence of skeleton joints. In [34], the skeleton joints are first transformed to a new coordinate system robust to translation, scale, and rotation; then a temporal sliding LSTM, including short-term, medium-term, and long-term components are applied. An average ensemble of these components is then used to find proper temporal features. Song et al. [41] presented an LSTM-based network to learn discriminative temporal and spatial information for activity recognition.

For CNN-based models, in order to satisfy the requirement for image data, joint coordinates are usually considered as pseudo-images so that convolution kernels can be applied to them. Du et al. [17] proposed to combine the joint positions and joint velocities, and used a two-stream CNN architecture; however, the long-term frame dependency is neglected. To overcome this drawback, Ke et al. [33] proposed to generate some clips out of videos that preserve temporal information and then feed them to a CNN-based network. In [37], a view-invariant recognition model is presented that finds an improved visualization of skeleton data and uses CNN as the classifier. In [49], co-occurrence features are found with a hierarchical framework which is effective for multisubject activities. Banerjee et al. [64] presented four representations, with complementary characteristics, for skeleton data and employed four complementary CNNs and a fuzzy fusion technique to combine their outputs and make the final decision.

Considering hinged joints and bones, the skeleton of human body can be modeled as a graph with nodes and vertices. There have been several successful graph-based methods proposed in the literature, leading this to a growing trend in the field [69]. A spatial-temporal graph convolutional network (GCN) is presented in [55], consisting of several spatial-temporal graph convolutions for extracting features of body skeleton. In [61], an attention-enhanced graph convolutional LSTM (AGC-LSTM) is employed for skeleton-based activity recognition. Shi et al. modeled and updated bones and joints employing directed graphs in [60]; The method is called directed graph neural network (DGNN). Cheng et al. [65] designed a decoupling GCN (DCGCN) to increase the ability of graph modeling for skeleton-based activity recognition, without adding extra computational costs. Shi et al. [68] proposed decoupled spatial-temporal attention

[1]Following the literature in the optimization field [5], we refer to the solution of our optimization problem as the optimal solution. We acknowledge that this may not be the true optimal solution as the optimization problem is nonconvex, and we can only find a local minimum point for the employed objective function, similar to other SOTA methods in the deep learning context.

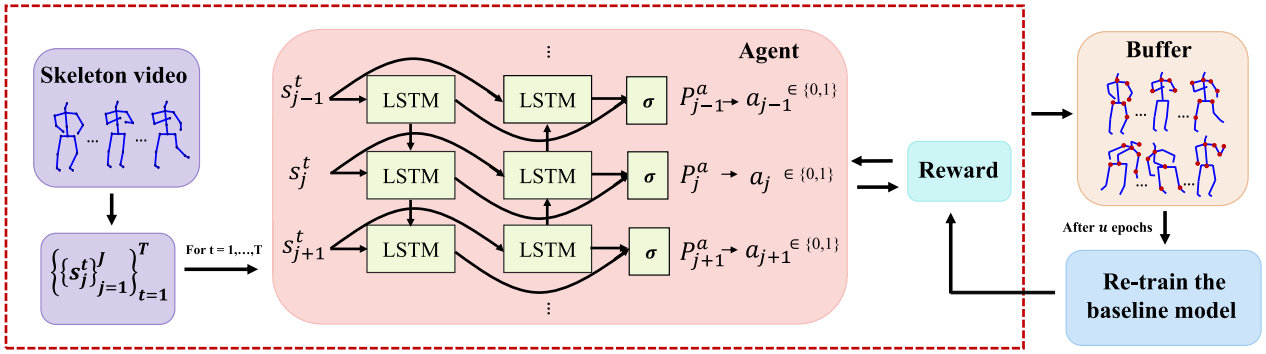

Fig. 2. Overall architecture of the proposed SHARL method. s_j^t is state of the jth joint in frame t of the input video and a_j is its corresponding action taken by the agent.

network to model spatial and temporal information, where both interframe and intraframe relationships between joints are considered.

All the skeleton-based activity recognition methods discussed above focus on designing novel deep architectures to learn discriminative spatial/temporal features. None of them are capable of being employed as a filtering block, to discard irrelevant joints, prior to the existing sophisticated deep learning-based skeleton-based recognition models. In this article, we approach this issue by proposing a novel spatial hard attention finding agent (filter) that discards the irrelevant joints and preserves the key ones, prior to the activity recognition in the testing phase.

B. Human Activity Recognition Using Reinforcement Learning

Reinforcement learning (RL) algorithms learn how to accomplish a complex objective by interacting with the environment, similar to the way human learns to act optimally in various environments [51]. In every RL algorithm, there exists an agent that explores the environment, which is usually expressed as a Markov decision process (MDP). The agent receives reward aligned with the agent's final goal(s), where the goal is to maximize an expected reward. DRL is an RL-based technique that employs deep neural networks to deal with high dimensional action/state spaces [31], [35], [47]. Deep RL has been a very successful technique and could achieve human-level performance in, e.g., Atari games [12], [20], [39], [59].

There are several studies in computer vision field where deep RL is used as the tool to solve problems, such as visual tracking [44], video captioning [53], action detection [29], person identification [25], and face recognition [40]. Moreover, [66] proposes a gesture recognition method using RL. However, there are few RL-based works for activity recognition, especially for skeleton data. Zhou et al. [56] developed an RL-based framework for selecting key frames in long RGB videos to summarize them. Chen et al. [45] proposed an activity recognition method, for RGB videos, that extracts features from different human body parts under a DRL framework. To find the most important frames in videos, multiagent RL is employed in [63], where each agent selects one frame at a time. An LSTM-based method developed by Dong et al. is presented in [58] which retrieves relevant frames based on RL. Both [58], [63] are designed for RGB video data. To the best of our knowledge, DPRL [52] is the only previous study that uses DRL for skeleton-based activity recognition. This method improves recognition performance by selecting key frames (i.e., finding hard temporal attentions) in skeleton videos, employing graph representation for the skeleton data, and a graph-based CNN for generating the required reward. Xu et al. [70] proposed a feature selection network (FSN) with actor-critic RL to select the most descriptive frames and discard ambiguous frames in a sequence. The features extracted for each frame of the skeleton sequence are generated by a generalized GCN (GGCN). The FSN contains a policy network and a value network which are actor and critic, respectively. Both networks are based on LSTM. In [72], an overview of using DRL techniques in the field of human activity recognition is presented. This article reviews some important deep RL methods and various aspects of deep RL-based human activity recognition techniques. It also covers different approaches for training and evaluating these models, along with their advantages and limitations. Interested readers may refer to this literature for deeper understanding of the field.

III. Proposed Method

Not all joints in a video are helpful for activity recognition. So it is necessary to identify and discard irrelevant joints to avoid the adverse effect of them. In the following, the proposed deep RL-based algorithm for finding spatial hard attentions, SHARL, is discussed.

A. Spatial-Attention-Aware Selection

The proposed SHARL method models the process of seeking the most discriminative joints as an MDP and solves it with the popular policy-based RL algorithm, Monte Carlo policy gradient (REINFORCE) [3]. Fig. 2 shows the overall architecture of SHARL. There is an agent in its current state of environment. The agent interacts with the environment through taking **actions**[2] that result in changing its state and receiving

[2]Two types of actions are used in this article: 1) the actions that should be recognized and 2) the **actions** in the RL framework. To avoid confusion, we use the bold-sized word to define the agent's **actions** in the RL framework.

reward. The agent learns to select discriminative joints, i.e., find spatial hard attentions, by maximizing the total expected reward.

In this article, the kth RL episode is denoted by $\mathcal{T}_k = (S_k, A_k, R_k)$, where S_k, A_k, and R_k are state, **action**, and reward at the kth episode, respectively. At each episode, the agent goes over all the T frames of a given video once, which means we have a single-step MDP. We defined single-step MDP to avoid the problem of delayed reward, which will be obtained only when the final set of selected joints are available [51]. One-step MDP is common in RL and policy gradient theorem can be proved for it. Moreover, as the input state of the agent in each frame of the video is different, the output of the agent can vary over frames of a single video and the whole resultant video is used to get the reward. We run the agent on the same video for K episodes; the set of episodes is denoted as $\mathcal{T} = (S_1, A_1, R_1, \ldots, S_K, A_K, R_K)$. Agent, state, **action**, and reward in the proposed RL-based selection process are as follows.

Agent: Any deep learning-based architecture (e.g., CNN-based, RNN-based, graph-based, etc.) can be used as the policy network. We can consider the human skeleton as an ordered sequence of J joints, where coordinate and motion of one joint may affect those of others. Bi-directional LSTM (BiLSTM) has shown its effectiveness when dealing with sequential data. Hence, as the agent, we use a BiLSTM-based network topped with a fully connected (FC) layer. At frame t of episode $\mathcal{T}_k$, the BiLSTM network gets the state S_k^t, defined below, as input and then feeds its output to the FC layer. The output of the agent is the probability vector $\{p_j^t\}_{j=1}^J$, which defines **actions** later.

State: Previous studies have shown that considering joints' motion, as well as their location, improves the activity recognition performance [61]. Therefore, we define the agent state at the t-th frame of $\mathcal{T}_k$ as $S_k^t = \{\mathbf{s}_j^t\}_{j=1}^J$ where $\mathbf{s}_j^t = [\mathbf{s}_{j,c}^t, \mathbf{s}_{j,m}^t]$; $\mathbf{s}_{j,c}^t$ is the 3-D coordinate of joint j and $\mathbf{s}_{j,m}^t$ is its corresponding 3-D motion vector, i.e., $\mathbf{s}_{j,m}^t = \mathbf{s}_{j,c}^t - \mathbf{s}_{j,c}^{t-1}$. The state set at the kth episode is $S_k = \{S_k^t\}_{t=1}^T$.

Action: The **action** is selection of a joint. We define two **actions** as "keep" and "remove." The output of the FC layer of the agent at frame t, $\{p_j^t\}_{j=1}^J$, denotes the probability of taking **action** "keep." Consider **action** set $A_k = \{\mathbf{a}_k^t\}_{t=1}^T$ where $\mathbf{a}_k^t$ is a J-dim indicator vector which shows the selected joints at frame t of the kth episode. If the jth element of $\mathbf{a}_k^t$, i.e., $a_{k,j}^t$ is 1, the jth joint is kept in frame t; otherwise it is removed. p_j^t indicates probability of setting $a_{k,j}^t$ to 1. All the J elements of $\mathbf{a}_k^t$ are sampled from Bernoulli distributions as follows:

$$\mathbf{a}_k^t = \left\{ a_{k,j}^t \sim \text{Bernoulli}\left(p_j^t\right) \right\}_{j=1}^J. \tag{1}$$

The joint selection process of the SHARL agent for a typical episode is depicted in Fig. 3.

Reward: The reward demonstrates the effectiveness of the agent's action, regarding the state. In our method, the reward is generated using a pretrained baseline recognition model which receives the T frames with the selected joints as input, where the selected joints are defined via **actions** that agent takes. We impose a strong punishment $-\Omega$, if the class label predicted by the baseline model changes from the correct label to a wrong one. If the turning goes otherwise, a strong reward of Ω is enforced. A reward value of r_0 is provided if no change is observed in the predicted class label, but the confidence of the baseline model toward predicting the correct class changes. r_0 is defined as follows:

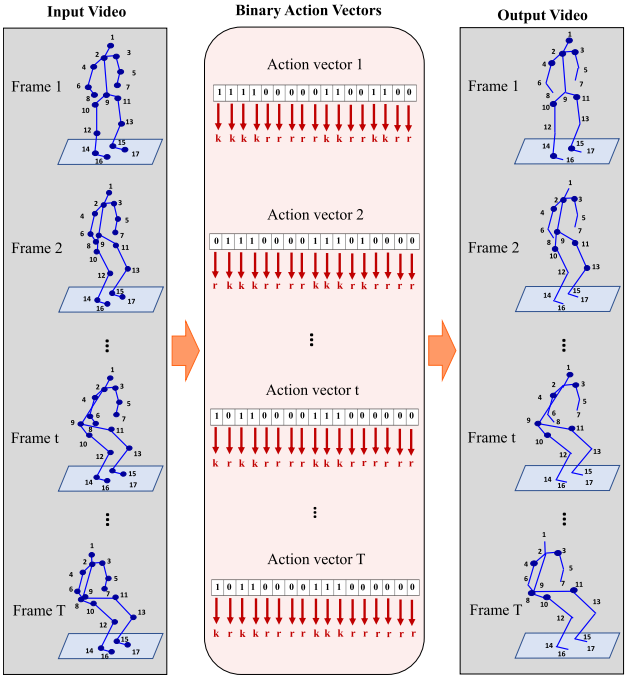

Fig. 3. Selection process for a typical episode with one step. The **action** arrows "k" and "r" mean the corresponding joints should be kept or removed, respectively.

$$r_0 = \text{sgn}\left(P_l^k - P_l^{k-1}\right) \tag{2}$$

where P_l^k is the probability of correctly classifying the video as class l in $\mathcal{T}_k$. The Reward at $\mathcal{T}_k$, i.e., R_k is shown as follows:

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

B. Training With REINFORCE

The goal of our spatial-attention-aware agent is to learn a policy function by maximizing the expected reward $\mathcal{R}(\theta)$ shown below

$$\mathcal{R}(\theta) = \text{E}_{p_\theta\left(a_{k,1:J}^{1:T}\right)}[R_k] \tag{4}$$

where $p_\theta(a_{k,1:J}^{1:T})$ is the probability distribution of the possible **actions** over the frames. We employ REINFORCE [3] to maximize the expected reward and find the corresponding optimal parameters θ.

REINFORCE is a well-known policy gradient approach, where the Monte Carlo strategy is employed such that a trajectory is sampled and the cumulative reward in every step of the trajectory is calculated. After that, the policy is updated based on the obtained rewards and this process is repeated until the optimal policy is found. The policy is approximated

by a function parameterized by θ. The gradient of the expected reward in REINFORCE can be shown as follows:

$$\nabla_\theta J(\theta) = \mathrm{E}_\pi[R_t \nabla_\theta \ln \pi_\theta(a_t|s_t)]. \tag{5}$$

Following the above equation, we compute the gradient of the expected reward, in the kth episode, with respect to the parameters θ as below

$$\nabla_\theta \mathcal{R}(\theta) = \mathrm{E}_{p_\theta\left(a^{1:T}_{k,1:J}\right)}\left[R_k \sum_{t=1}^{T}\sum_{j=1}^{J} \nabla_\theta \ln \pi_\theta\left(a^t_{k,j}|s^t_{k,j}\right)\right] \tag{6}$$

where π_θ is the policy function, and $s^t_{k,j}$ is s^t_j at the kth episode. As discussed before, we run the agent for K episodes on each input skeleton video. Hence, we use the average gradient shown in (7) when training the agent

$$\nabla_\theta \mathcal{R}(\theta) \approx \frac{1}{KT}\sum_{k=1}^{K}\left[R_k \sum_{t=1}^{T}\sum_{j=1}^{J} \nabla_\theta \ln \pi_\theta\left(a^t_{k,j}|s^t_{k,j}\right)\right]. \tag{7}$$

To reduce variance in training θ and to improve convergence, we normalize the reward by subtracting a constant baseline b, where b is the average reward of episodes. Therefore, the gradient becomes

$$\nabla_\theta \mathcal{R}(\theta) \approx \frac{1}{KT}\sum_{k=1}^{K}\left[(R_k - b) \sum_{t=1}^{T}\sum_{j=1}^{J} \nabla_\theta \ln \pi_\theta\left(a^t_{k,j}|s^t_{k,j}\right)\right]. \tag{8}$$

We consider some more terms in our objective function, besides maximizing the expected reward $\mathcal{R}(\theta)$. We would like to limit the maximum number of selected joints to a user-settable value N, where N is an integer value between 1 and J; this can be realized by considering $\mathbf{1}^T\mathbf{p} \leq N$ as a constraint of the optimization problem. In addition, we would like to encourage the agent to keep at least one joint per frame; this can be realized by considering $\mathbf{1}^T\mathbf{p} \geq 1$ as a constraint of the optimization problem. In this article, to be able to solve the optimization problem using stochastic gradient descent, the effect of these two additional constraints are considered, respectively, in the second and third terms of the final objective function shown below

$$\min_\theta -\mathcal{R}(\theta) + \alpha \times \left(\mathbf{1}^T\mathbf{p} - N\right) - \beta \times \left(\mathbf{1}^T\mathbf{p}\right) \tag{9}$$

where $\mathbf{p}$ denotes the average probability of **action** vectors over the T frames, and α and β are two hyper-parameters to control contribution of their corresponding terms.

C. Retraining Baseline

Although we can use a pretrained baseline model with frozen parameters for the reward generation and video classification, the performance of the baseline can further be improved if we fine-tune its parameters considering the selected joints. We frequently update the baseline model parameters when training the proposed spatial-attention-aware agent. The update interval u changes during the training phase according to the agent's expertise. More specifically, at the beginning, as the agent is still unskilled, the baseline is updated after a relatively long interval, i.e., after passing over half of the epochs; the update interval is then decreased with factor 2. A buffer, with size s, is used for baseline updating which is filled with the latest s video outputs of agent, i.e., the latest s videos with selected joints. The baseline model is updated every u epochs using the videos stored in the buffer. Our experiments show that employing such a buffer makes the baseline retraining phase faster and more efficient.

Algorithm 1 Proposed SHARL Method

Input: The video sequences with labels, epochs, K
Output: Trained spatial-attention-aware agent

```
Initialization: pre-train the baseline model, randomly initialize the agent network, count = 0, buffer = [], u = number of epochs / 2
for epochs do
    count += 1
    for videos do
        for K episodes do
            run the policy network
            find the action using (1)
            take the action and update the state
            compute reward using (2) and (3)
        end for
        compute the average reward
        compute the loss (9)
        update the agent network parameters
        update the buffer
        if count = u then
            retrain the baseline model using the buffer
            count = 0
            u = ⌈u/2⌉
        end if
    end for
end for
```

Pseudo-code of SHARL is presented in Algorithm 1. In brief, the baseline model is first pretrained by the original training data (with full set of joints). A sequence of video is then given to the agent, K episodes are completed, joints are selected, and the policy network is updated. We repeat this process for u epochs, and then retrain the classifier using the videos (with subset of selected joints) stored in the buffer. The procedure is repeated for all epochs.

IV. Experimental Results

To analyze the performance of our proposed SHARL method, we conducted experiments on three benchmark activity recognition datasets. Effectiveness of the proposed spatial-attention-aware agent in improving the baseline model performance is demonstrated on four different models, selected from all the three activity recognition model categories discussed in Section II-A, ranging from simple baselines (i.e., CNN- and LSTM-based models) to the recent advanced graph-based models (i.e., DGNN [60] and DCGCN [65]). The recognition performance of SHARL (with DGNN as baseline) is compared to several state-of-the-art methods on skeleton-based activity recognition. This section is organized as follows. Datasets description is presented in Section IV-A. The employed network architectures and hyperparameters in SHARL are presented in Section IV-B. Description of the baseline models and the SHARL performance with different

baselines are presented in Section IV-C. In Section IV-D, we compare the performance of SHARL with several state-of-the-art skeleton-based recognition methods. The learned hard attention is visualized in Section IV-E. Analysis of SHARL convergence, the effect of hyperparameters α and β, and sensitivity of the algorithm to hyperparameter N are discussed in Sections IV-F, IV-G, and IV-H, respectively. In the end, in Section IV-I, the effect of SHARL on training run time is investigated.

A. Datasets

NTU+RGBD Dataset (NTU): Having 56 880 sequences and 4 million frames, NTU is currently the largest available dataset for activity recognition [27]. The video samples were captured from 40 different human subjects and belonged to 60 different activity classes. For train/test partitioning, NTU has two settings: 1) cross-subject (CS) and 2) cross-view (CV). In the CS setting, 40 320 samples captured from 20 subjects are considered as training samples and the other 16 540 samples are used as test samples. In the CV setting, 37 920 samples of camera views 2 and 3 are included in the training set and the remaining samples that are captured by the other camera, i.e., camera 1, is used as the test set. There are either one or two persons in each video, and the information on the 25 joints is recorded for each person.

SBU Kinect Interaction Dataset (SBU): It includes 230 sequences of 6614 frames [10]. The samples belong to eight classes and all the labels are two-person interactive activities. The data has a fivefold cross-validation setting. The number of skeleton joints is 15 for each subject; therefore, there are 30 joints in each frame.

UT-Kinect Dataset (UT): It has 200 video sequences belonging to ten classes of activities and each of them is performed by ten subjects two times [9]. To evaluate SHARL on this dataset, Leave-one-out cross-validation protocol is used. There is no interactive activity in this data, i.e., all samples have one subject. Each subject is represented by 20 joints.

B. Implementation Details

A 3-layer BiLSTM is employed as the agent's network (policy network), Adam is the used optimizer with an initial learning rate 5e-3 and the dropout rate is set to 0.5. Hyperparameters K, Ω, α, and β are, respectively, set to 4, 10, 0.01, and 0.009. We set hyperparameter N to half of the number of all the available joints. More specifically, N is set to $\lceil J/2 \rceil$ where $\lceil . \rceil$ denotes the ceiling function. The buffer size s is set to 25. The number of epochs can be selected adaptively based on the value of the loss function defined in (9). However, for simplicity, it is set to 25 in our experiments on all datasets. The proposed method is implemented in python using the deep-learning framework Pytorch.

C. Boosting Baseline

The proposed spatial-attention-aware agent improves the performance of the employed baseline recognition model. To demonstrate this, we build and train SHARL using four different baseline models and compare the recognition performance with and without the joint selection performed by the agent. The employed baselines include both simple models, i.e., CNN-based and BiLSTM-based, and complex ones, i.e., graph-based networks DCGCN and DGNN.

TABLE I
ACCURACY RESULTS (IN PERCENT) OF FOUR DIFFERENT BASELINE MODELS WITH AND WITHOUT SHARL

Method	CS	CV	SBU	UT	avg.
BiLSTM	65.0	69.2	76.0	94.5	76.17
CNN	70.2	71.3	78.2	87.9	76.9
DCGCN	88.1	95.2	88.2	98.2	92.1
DGNN	89.9	96.1	87.8	97.5	92.5
SHARL-BiLSTM	**68.5**	**71.9**	**81.1**	**97.9**	**84.15**
SHARL-CNN	**76.1**	**81.8**	**85.5**	**96.2**	**82.15**
SHARL-DCGCN	**89.3**	**96.2**	**88.6**	**99.1**	**93.07**
SHARL-DGNN	**90.4**	**96.5**	**88.7**	**98.9**	**93.1**

BiLSTM, which is an extension of a simple LSTM, is mainly used for classification of sequential data, and includes one forward and one backward LSTM to provide more content for the network and make the training procedure faster and more effective. LSTM has memory cell, and input and forget gates. The input gate and forget gate control the flow of information into the memory cell which mitigates the gradient vanishing and exploding problem. See [4], [24] for more details. CNNs are another class of deep learning methods that are mostly used for image and video data as they can effectively find the spatial and temporal dependencies through applying some filters. In such networks, there are several convolution layers topped with FC layers at the end. Due to using different filters, CNNs can find shift invariant or space invariant representations for their input data. More details can be found in [2] and [24]. GCNs are designed for data with graph structure, and similar to CNNs, have some shared filter parameters over the graph vertices. DGNN [60] is specifically developed for skeleton-based activity recognition using directed acyclic graphs which are proper for skeleton modeling. DGNN incorporates both spatial and motion information in a two-stream structure. DCGCN is another recent GCN-based model that employs a decoupling graph convolution whereby different channels have different independent trainable adjacent matrix. It also uses a drop-graph module for regularization [65].

The CNN-based recognition model we used has two convolution layers followed by one FC layer, and is trained using the Adam optimization method. The BiLSTM model has three layers with hidden layer size 256, and the optimizer is Adam. In our experiments, we use the DCGCN and DGNN codes available on the respective author websites. For a fair comparison, the default settings for each algorithm suggested in the original papers are used. The hyperparameters of our SHARL method are also set to their default values presented in Section IV-B. Recognition performance of the baseline models and the proposed SHARL method with different baselines, on the benchmark datasets explained above, are shown in Table I. SHARL-X denotes the proposed SHARL model with baseline model X. On each dataset, among X and

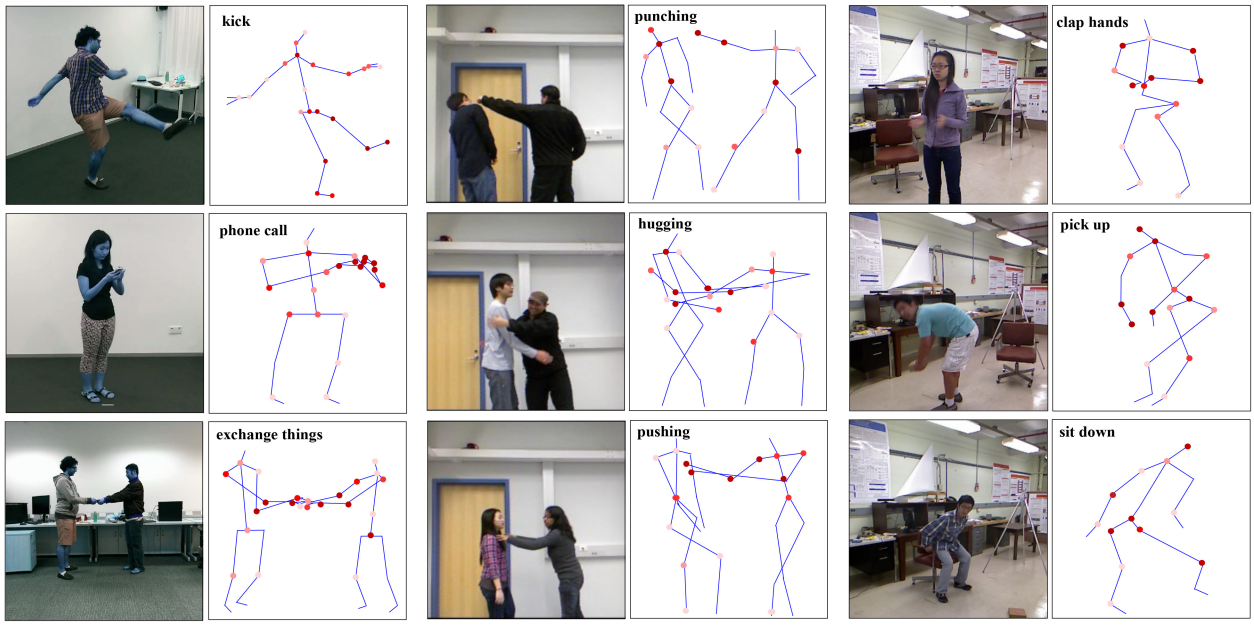

Fig. 4. Visualizing frequency of selecting a joint by SHARL within a video, for three different activity classes from NTU, SBU, and UT datasets. The higher the red color intensity, the higher is the selection frequency. (a) Three activities of NTU dataset. (b) Three activities of SBU dataset. (c) Three activities of UT dataset.

TABLE II
ACCURACY RESULTS (IN PERCENT) OF DIFFERENT METHODS ON NTU DATASET

Method	CS	CV	year
Lie Group [16]	50.1	52.8	2014
HBRNN[18]	59.1	64.0	2015
Part-aware LSTM [27]	62.9	70.3	2016
Mengyuan et al. [37]	76	82.56	2017
LieNet-3Blocks [32]	61.4	67.0	2017
DPRL+GCNN [52]	83.5	89.8	2018
ST-GCN [55]	81.5	88.3	2018
SA-LSTM [50]	71.9	80.4	2018
DGNN [60]	89.9	96.1	2019
AGC-LSTM [61]	89.2	95.0	2019
DCGCN [65]	88.1	95.2	2020
SHARL	**90.4**	**96.5**	

TABLE III
ACCURACY RESULTS (IN PERCENT) OF DIFFERENT METHODS ON SBU DATASET

Method	SBU	year
Raw skeleton[11]	49.7	2012
Joint feature[14]	86.9	2014
CHARM [19]	83.9	2015
Hierarchical RNN[18]	80.35	2015
SA-LSTM [50]	88.0	2018
DGNN [60]	87.8	2019
DCGCN [65]	88.2	2020
SHARL	**88.7**	

SHARL-X, the best one is shown in bold. As can be seen, the proposed spatial-attention-aware selection agent improves the recognition accuracy of all the baseline models. The last column shows recognition accuracy averaged over all datasets. This column indicates that the proposed SHARL framework significantly enhances the performance of the baseline models.

D. Comparison to State-of-the-Art

In this section, we adopt DGNN as the SHARL baseline model. SHARL (with DGNN) is compared with several state-of-the-art skeleton-based methods. Accuracy comparison on the two subcategories of NTU dataset, i.e., CS and CV, is shown in Table II. Among the eleven activity recognition algorithms, the proposed SHARL method yields the best accuracy. A comparison on the SBU dataset is shown in Table III. As can be seen, the proposed method outperforms all the other state-of-the-art methods on this dataset. Performance comparison with state-of-the-art on UT dataset is presented in Table IV. Among the eleven algorithms, SHARL provides the best performance.

E. Visualization of the Learned Hard Attention

The joints selected by the proposed spatial-attention-aware agent, for three different activities from the three benchmark datasets, are visualized in Fig. 4. The red circle color brightness at each joint shows the selection frequency of that joint across the whole video frames; e.g., in the "phone call" activity of the NTU dataset, hand-related joints are correctly selected in all frames and the redundant joints, e.g., in feet and head are correctly removed. Fig. 5 demonstrates distribution of the selected joints over all the activities for each data set. As can be seen, the selected joints vary among different activities. Also, depending on the activity, either the upper body or lower body skeleton joints are mainly selected. These experiments

TABLE IV
ACCURACY RESULTS (IN PERCENT) OF DIFFERENT METHODS ON UT DATASET

Method	UT	year
Histogram of 3D Joints [9]	90.9	2012
Grassmann Manifold [22]	88.5	2015
Riemannian Manifold [13]	91.5	2015
SCK+DCK [26]	98.2	2016
GMSM [28]	97.4	2016
ST-NBNN [43]	98.0	2017
ST-LSTM+Trust Gate [36]	97.0	2017
DPRL+GCNN [52]	98.5	2018
DGNN [60]	97.5	2019
DCGCN [65]	98.2	2020
SHARL	**98.9**	

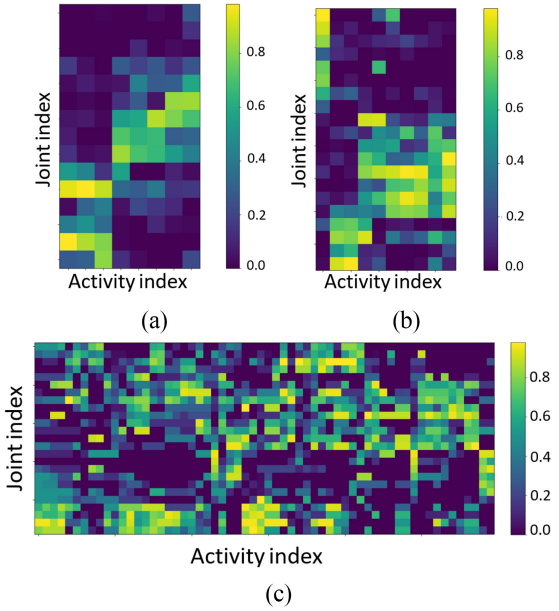

Fig. 5. Distribution of the engaged joints in different actions for (a) SBU, (b) UT, and (c) NTU(CS) dataset.

confirm that the learned hard attention (i.e., selected joints) are consistent with what human perceives.

F. Convergence and Stability

As is discussed in Section III, the proposed agent is trained by minimizing the loss function defined in (9). Based on [51], policy gradients are guaranteed to converge to at least a local optimum. To illustrate it in our experiments, the loss value versus training iteration, for datasets SUB, UT, and NTU (CS), are shown in Fig. 6. These graphs show that, at the training outset, the loss oscillates significantly. This can be associated with the low skill level of the agent in finding hard attentions (i.e., relevant joints per frame). As the training phase progresses, the agent gains more and more skills in identifying and selecting relevant joints; hence, as is desired, the loss converges to a very small value at the end of training. Moreover, in our experiments, we observe that the SHARL performance does not vary significantly in different executions of the algorithm, which is a desired property that hints the algorithm is stable. This property is quantified in Fig. 7, where the confidence interval[3] of loss during iterations for five different executions of SHARL on one fold of SBU is shown, where BiLSTM is used as baseline. As can be seen, the variance of output losses is high at the beginning, but it turns to very small numbers at the end of the training, the time that algorithm converges. This implies the stability of SHARL.

G. Influence of Hyperparameters α and β

In this section, the effects of hyperparameters α and β in (9) are investigated. Fig. 8(a) shows activity recognition accuracy where $\alpha \in \{0, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$, and β is set to its default value, i.e., 9^{-3}. Fig. 8(b) shows the recognition accuracy where $\beta \in \{0, 9^{-4}, 9^{-3}, 9^{-2}, 1\}$ and α is set to its default value 10^{-2}. In both of these experiments, N is set to its default value $\lceil J/2 \rceil$. SHARL-BiLSTM and the UT dataset are used for this experiment. The graphs confirm the effectiveness of including these two additional constraints in the optimization problem. α and β control the effect of the upper bound and the lower bound we considered for the number of selected joints in the loss function. Fig. 8(a) shows the best range for α is around 10^{-2} and 10^{-1}. By further increasing α, the algorithm focuses more on minimizing the second component of (9), i.e., selecting no more than N joints, and pays less attention to other components, resulting in accuracy degradation. Fig. 8(b) shows the best recognition performance is achieved when β is in the range of 9^{-4} to 9^{-3}. It can be seen that SHARL achieves high performance for a wide range of α and β, which is a desirable property.

H. Sensitivity to Hyperparameter N

In this section, we investigate the effect of changing N on the performance of SHARL (with BiLSTM baseline). Fig. 9 shows the accuracy versus N for all the data sets, where $N \in \{3, 7, 10, 15, 19, 26, 30\}$, $N \in \{2, 5, 8, 10, 13, 17, 20\}$, and $N \in \{5, 10, 15, 25, 30, 40, 50\}$, respectively, for the SBU, UT, and NTU datasets. Note J is 30, 20, and 50 in SBU, UT, and NTU datasets. The role of N in (9) is imposing an upper limit on the number of selected joints. Fig. 9 shows that by increasing N to half of the number of joints, the SHARL performance significantly improves. The method maintains high accuracy for a wide range of N. This demonstrates that, as is desired, the SHARL method is not too sensitive to the user-settable parameter N and that the method inherently tends to select only relevant joints. Accuracy is computed using fivefold and leave-one-out cross-validation procedures for the SBU and UT datasets, respectively.

I. Run Time Improvement

Similar to most machine learning algorithms, the SHARL method is trained offline and the test phase, where joints are selected, can be done in real time. Our experiments show that, on average, the proposed SHARL framework eliminates

[3]Confidence interval denotes the mean of an estimation plus/minus its variance.

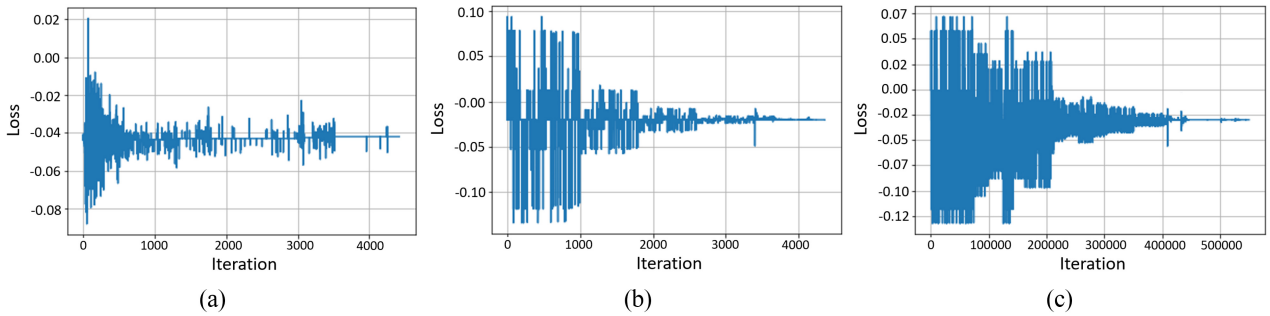

Fig. 6. SHARL-BiLSTM loss versus iterations for (a) SBU, (b) UT, and (c) NTU(CS) dataset.

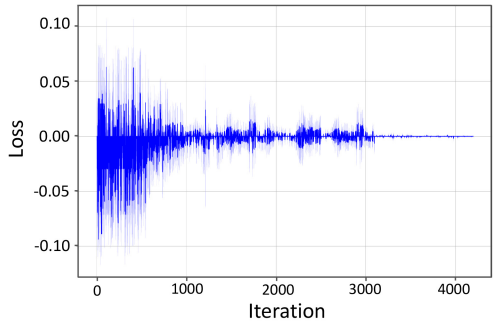

Fig. 7. Confidence interval of loss during iterations for five different runs of the SHARL method on one fold of SBU, and using BiLSTM as baseline.

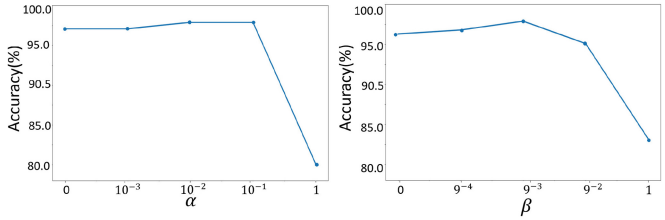

Fig. 8. Accuracy of SHARL-BiLSTM versus (a) α and (b) β for UT dataset.

TABLE V
RUNTIME (IN HOURS) OF TRAINING A BILSTM-BASED RECOGNITION MODEL WITH (T_{pr+p}) AND WITHOUT (T_o) SHARL

Method	CS	CV	SBU	UT	avg.
T_{pr}	3.51	3.32	0.01	0.01	1.71
T_p	136.45	130.36	0.47	0.55	66.95
T_{pr+p}	139.96	133.68	0.49	0.57	68.67
T_o	150.70	145.08	0.68	0.61	74.26

about 60% of joints. One may employ the trained spatial-attention-aware agent as a preprocessing (filter) block before (re-)training a recognition model. This may decrease the (re-)training phase time since only relevant joints are involved. This may be useful in online activity recognition applications when a faster yet effective (re-)training phase is appreciated. To demonstrate this property, a pretrained SHARL agent (trained with BiLSTM baseline) is applied to the data and relevant joints are identified and selected. The required time for this preprocessing phase, for each dataset, is recorded and shown by T_{pr} in Table V. The preprocessed videos, in which the irrelevant joints are removed, are then used to train a BiLSTM-based recognition model, where weights are randomly initialized. The required time to train the BiLSTM-based recognition model with preprocessed videos is recorded and shown by T_p in the table. The total required time $T_{pr+p} = T_p + T_{pr}$ is reported in the third row of the table. The required time to train the BiLSTM-based recognition model using the *original* full-joint videos, with random initial weights, is also recorded and shown by T_o. The same number of training epochs are used when measuring T_p and T_o. The average run times (avg.) for all the four datasets are shown in the last column of the table. The run times reported in the table are in hours. Comparing T_{pr+p} with T_o shows that employing the pretrained SHARL agent as a preprocessing block speeds up the training phase of the recognition model on average by about 8%. In addition, we observed that training of the recognition model converges faster, i.e., requires less number of training epochs, when using the preprocessed data as input. One Tesla P100-PCIE-16-GB GPU is used to run these experiments.

V. CONCLUSION AND DISCUSSION

In this article, we discovered a novel problem for activity recognition, that is irrelevant joints should be identified and discarded to improve recognition performance. We proposed a novel spatial hard attention finding method, SHARL, employing deep RL without requiring extra labels. We formulated the process of mining the key joints as an MDP and found the optimal solution using REINFORCE. SHARL associates each video frame with its own optimal joint set, which can vary both in size and membership across the video. This allows to incorporate in recognition the fact that the important key joints may change as an activity proceeds. The proposed SHARL framework is extensible—i.e., it can be applied to the existing deep learning-based activity recognition models to improve their performance. Performance of the proposed framework is demonstrated on three widely used benchmark activity recognition datasets NTU, SBU, and UT-kinect using four different baseline models. Our method achieved a very competitive performance of activity recognition compared to state-of-the-art skeleton-based activity recognition methods. Experimental results show that SHARL can increase the training speed of a recognition model, when employed as a selection block before the recognition model.

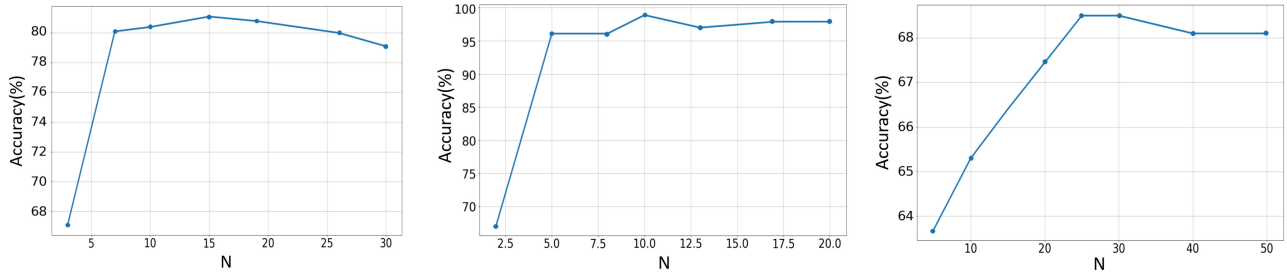

Fig. 9. Accuracy of SHARL-BiLSTM versus N for (a) SBU, (b) UT, and (c) NTU(CS) datasets.

In general, selecting relevant joints of a human body in activity recognition can be beneficial in several ways.

1) *Improved Accuracy:* By selecting only the relevant joints, the model's performance can be improved, as it can focus on the most informative parts of the body. This can lead to more accurate predictions of activities.
2) *Reduced Computational Complexity:* By selecting only the relevant joints, the computational complexity of the model can be reduced. This can lead to faster predictions and reduced resource usage, making the model more efficient.
3) *Reduced Noise:* By excluding irrelevant joints, the noise in the data can be reduced, leading to a more reliable and robust model.
4) *Improved Interpretability:* By focusing on only the relevant joints, it becomes easier to understand which parts of the body are most important for each activity, making the model more interpretable.
5) *Generalization:* A model trained on a subset of the joints may generalize better to new data, as it is more focused on the most informative parts of the body.

Overall, selecting relevant joints for activity recognition can lead to better performance, improved efficiency, and increased interpretability.

ACKNOWLEDGMENT

The authors would like to thank the Natural Sciences and Engineering Research Council of Canada and the Department of Electrical and Computer Engineering at McGill University. This work would not have been possible without their financial support. The authors would also like to thank Calcul Quebec and Compute Canada for providing the necessary computational resources to conduct our experiments.

REFERENCES

[1] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Percept. Psychophys.*, vol. 14, no. 2, pp. 201–211, 1973.
[2] W. Zhang, J. Tanida, K. Itoh, and Y. Ichioka, "Shift-invariant pattern recognition neural network and its optical architecture," in *Proc. Annu. Conf. Japan Soc. Appl. Phys.*, Montreal, QC, Canada, 1988, pp. 2147–2151.
[3] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, 1992.
[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[5] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[6] T. Theodoridis and H. Hu, "Action classification of 3D human models using dynamic ANNs for mobile robot surveillance," in *Proc. IEEE Int. Conf. Robot. Biomimet. (ROBIO)*, 2007, pp. 371–376.
[7] Z. Ren, J. Meng, J. Yuan, and Z. Zhang, "Robust hand gesture recognition with Kinect sensor," in *Proc. 19th ACM Int. Conf. Multimedia*, 2011, pp. 759–760.
[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.
[9] L. Xia, C.-C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3D joints," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 20–27.
[10] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras, "Two-person interaction detection using body-pose features and multiple instance learning," in *Proc. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2012, pp. 28–35.
[11] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras, "Two-person interaction detection using body-pose features and multiple instance learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 28–35.
[12] V. Mnih et al., "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.
[13] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. De. Bimbo, "3D human action recognition by shape analysis of motion trajectories on Riemannian manifold," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1340–1352, Jul. 2015.
[14] Y. Ji, G. Ye, and H. Cheng, "Interactive body part contrast mining for human interaction recognition," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, 2014, pp. 1–6.
[15] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," 2014, *arXiv:1406.2199*.
[16] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 588–595.
[17] Y. Du, Y. Fu, and L. Wang, "Skeleton based action recognition with convolutional neural network," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, 2015, pp. 579–583.
[18] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1110–1118.
[19] W. Li, L. Wen, M. C. Chuah, and S. Lyu, "Category-blind human action recognition: A practical recognition system," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4444–4452.
[20] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[21] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," in *Proc. Neural Inf. Process. Syst. Time Series Workshop*, 2015, pp. 1–11.
[22] R. Slama, H. Wannous, M. Daoudi, and A. Srivastava, "Accurate 3D action recognition using learning on the Grassmann manifold," *Pattern Recognit.*, vol. 48, no. 2, pp. 556–567, 2015.
[23] Y. Du, Y. Fu, and L. Wang, "Representation learning of temporal dynamics for skeleton-based action recognition," *IEEE Trans. Image Process.*, vol. 25, pp. 3010–3022, 2016.
[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[25] A. Haque, A. Alahi, and L. Fei-Fei, "Recurrent attention models for depth-based person identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1229–1238.

[26] P. Koniusz, A. Cherian, and F. Porikli, "Tensor representations via kernel linearization for action recognition from 3D skeletons," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 37–53.

[27] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1010–1019.

[28] P. Wang, C. Yuan, W. Hu, B. Li, and Y. Zhang, "Graph based skeleton motion representation and similarity measurement for action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 370–385.

[29] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2678–2687.

[30] W. Zhu et al., "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, 2016, pp. 1–7.

[31] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[32] Z. Huang, C. Wan, T. Probst, and L. Van Gool, "Deep learning on Lie groups for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6099–6108.

[33] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3288–3297.

[34] I. Lee, D. Kim, S. Kang, and S. Lee, "Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1012–1020.

[35] Y. Li, "Deep reinforcement learning: An overview," 2017, *arXiv:1701.07274*.

[36] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang, "Skeleton-based action recognition using spatio-temporal LSTM network with trust gates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3007–3021, Dec. 2018.

[37] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognit.*, vol. 68, pp. 346–362, Aug. 2017.

[38] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3D human pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2640–2649.

[39] N. D. Nguyen, T. Nguyen, and S. Nahavandi, "System design perspective for human-level agents using deep reinforcement learning: A survey," *IEEE Access*, vol. 5, pp. 27091–27102, 2017.

[40] Y. Rao, J. Lu, and J. Zhou, "Attention-aware deep reinforcement learning for video face recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3931–3940.

[41] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, 2017, pp. 1–8.

[42] H. Wang and L. Wang, "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 499–508.

[43] J. Weng, C. Weng, and J. Yuan, "Spatio-temporal Naive-Bayes nearest-neighbor (ST-NBNN) for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4171–4180.

[44] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2711–2720.

[45] L. Chen, J. Lu, Z. Song, and J. Zhou, "Part-activated deep reinforcement learning for action prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 421–436.

[46] G. Ciocca, A. Elmi, P. Napoletano, and R. Schettini, "Activity monitoring from RGB input for indoor action recognition systems," in *Proc. IEEE 8th Int. Conf. Consum. Electron. (ICCE)*, 2018, pp. 1–4.

[47] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," 2018, *arXiv:1811.12560*.

[48] R. A. Güler, N. Neverova, and I. Kokkinos, "DensePose: Dense human pose estimation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7297–7306.

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

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

[51] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[52] Y. Tang, Y. Tian, J. Lu, P. Li, and J. Zhou, "Deep progressive reinforcement learning for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5323–5332.

[53] X. Wang, W. Chen, J. Wu, Y.-F. Wang, and W. Y. Wang, "Video captioning via hierarchical reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4213–4222.

[54] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 466–481.

[55] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 1–10.

[56] K. Zhou, Y. Qiao, and T. Xiang, "Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 1–9.

[57] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.

[58] W. Dong, Z. Zhang, and T. Tan, "Attention-aware sampling via deep reinforcement learning for action recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8247–8254.

[59] M. Jaderberg et al., "Human-level performance in 3D multiplayer games with population-based reinforcement learning," *Science*, vol. 364, no. 6443, pp. 859–865, 2019.

[60] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7912–7921.

[61] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional LSTM network for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1227–1236.

[62] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5693–5703.

[63] W. Wu, D. He, X. Tan, S. Chen, and S. Wen, "Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6222–6231.

[64] A. Banerjee, P. K. Singh, and R. Sarkar, "Fuzzy integral based CNN classifier fusion for 3D skeleton action recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 6, pp. 2206–2216, Jun. 2021.

[65] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, and H. Lu, "Decoupling GCN with dropgraph module for skeleton-based action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 536–553.

[66] X. Gao, Y. Jin, Q. Dou, and P.-A. Heng, "Automatic gesture recognition in robot-assisted surgery with reinforcement learning and tree search," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2020, pp. 8440–8446.

[67] B. Ren, M. Liu, R. Ding, and H. Liu, "A survey on 3D skeleton-based action recognition using learning method," 2020, *arXiv:2002.05907*.

[68] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 1–16.

[69] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with multi-stream adaptive graph convolutional networks," *IEEE Trans. Image Process.*, vol. 29, pp. 9532–9545, 2020.

[70] Z. Xu, Y. Wang, J. Jiang, J. Yao, and L. Li, "Adaptive feature selection with reinforcement learning for skeleton-based action recognition," *IEEE Access*, vol. 8, pp. 213038–213051, 2020.

[71] B. Nikpour and N. Armanfard. "Joint selection using deep reinforcement learning for skeleton-based activity recognition." TechRxiv. 2021. [Online]. Available: https://doi.org/10.36227/techrxiv.14887869.v1

[72] B. Nikpour, D. Sinodinos, and N. Armanfard. "Deep reinforcement learning in human activity recognition: A survey." TechRxiv. 2022. [Online]. Available: https://doi.org/10.36227/techrxiv.19172369.v3