Two-Person Mutual Action Recognition Using Joint Dynamics and Coordinate Transformation

Shian-Yu Chiu[1], Kun-Ru Wu[2], Yu-Chee Tseng[3,4,5]
{jimmychiu702@gmail.com[1], wufish@nycu.edu.tw[2], yctseng@nycu.edu.tw[3,4,5] }

Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, 30010, Taiwan[1,2],
College of AI, National Yang Ming Chiao Tung University, Hsinchu, 30010, Taiwan[3],
Academia Sinica, Taipei, 11529, Taiwan[4],
Kaohsiung Medical University, Kaohsiung, 80708, Taiwan[5]

Abstract. Skeleton-based action recognition has attracted lots of attention in computer vision. Human mutual interaction recognition relies on extracting discriminative features for better understanding details. In this work, we propose two vectors to encode joint dynamics and spatial interaction information. The proposed model shows remarkable performance at handling sequential data. Experimental results demonstrate that our model outperforms state-of-the-art approaches with much less overheads.

Keywords: Skeleton-based Action Recognition, Mutual Interaction Recognition, Bidirectional LSTM, Deep Learning, Human Behavior Analysis.

1 Introduction

Human action recognition has been an active research topic since there are a variety of applications, such as video surveillance, video understanding, and human-computer interaction [1, 2, 3]. A lot of researches have focused on single-person action recognition [4, 5, 6, 7, 8, 9, 10, 11, 12]; however, lots of social actions occur between/among persons. Mutual action recognition has been studied in computer vision [13, 14, 15, 16, 17, 18]. In this task, extracting representative spatial-temporal information within mutual interactions is crucial for understanding the relationship of joint movements conducted by two persons.

Most of the conventional studies focus on recognizing actions from RGB videos recorded by 2D cameras [19, 20, 21, 22]. However, there remain three main challenging problems not fully addressed. First, it loses depth information from the 3D space. Second, the recognition results are highly vulnerable to recording distance and angle, human body occlusion, and background changes. Third, it is difficult to extract useful features, like human poses and relationship between joints, from high dimensional input data directly.

CAIP 2021, November 20-24, Bologna, Italy

DOI 10.4108/eai.20-11-2021.2314154

With the development of depth sensors, such as Intel RealSense [23] and Microsoft Kinetic [24], and the advance of human pose estimation [25, 26, 27, 28], skeleton data is easily accessible and several approaches based such data have been proposed. Human actions can be represented as a series of articulated skeleton frames, which are more robust to the variations of background and viewpoint changes. Recently, Graph Convolutional Networks (GCNs) has also been applied to skeleton structures for action recognition [11, 29, 30, 12] and it has demonstrated promising performance.

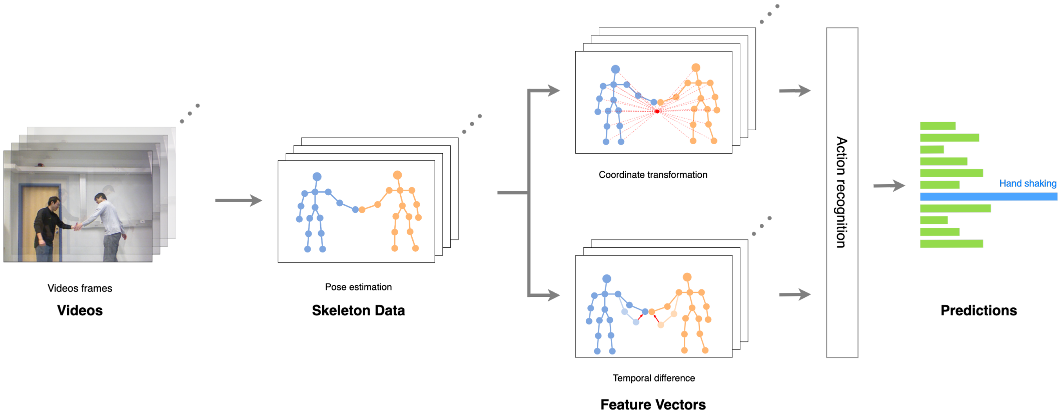


Fig. 1. Workflow of the skeleton-based interaction recognition.

In skeleton-based action recognition, an action is represented as a sequential 3-dimensional positions of joints. However, in this representation, there are two potential problems. First, using only positions cannot explicitly encode the joint dynamics. Second, the coordinates of joints are sensitive to camera setup and human locations. Thus, more complicated networks, such as GCNs, are required to explore more informative features, and it leads to more computational complexity. We propose to construct two feature vectors, *Coordinate Transformation Vectors* (CTVs) and *Temporal Difference Vectors* (TDVs). CTVs can mitigate the impact of camera setup difference and obtain more spatial interaction information. TDVs can encode the joint dynamics throughout a frame sequence. In addition, the low-dimensional but more informative feature vectors allow us to adopt a more lightweight network than the state-of-the-arts models and speed up training and inferring processes. The generated features are concatenated and fed into a stacked bidirectional Long Short-Term Memory (LSTM) network, which explores deeper forward and backward information. Fig. 1 illustrates the whole process of the skeleton-based mutual action recognition. To validate our claims, we evaluate our model on three datasets, SBU [13], NTU RGB+D [31], and NTU RGB+D 120 [32]. Our model outperforms the state-of-the-art approaches on these datasets with much less computation. The floating-point operations per second (FLOPs) v.s. accuracy diagram on NTU RGB+D is shown in Fig. 2.

To summarize the contributions of this work, we propose two skeleton data representations (CTVs and TDVs) to encode joint dynamics and spatial interaction information, a stacked bidi-

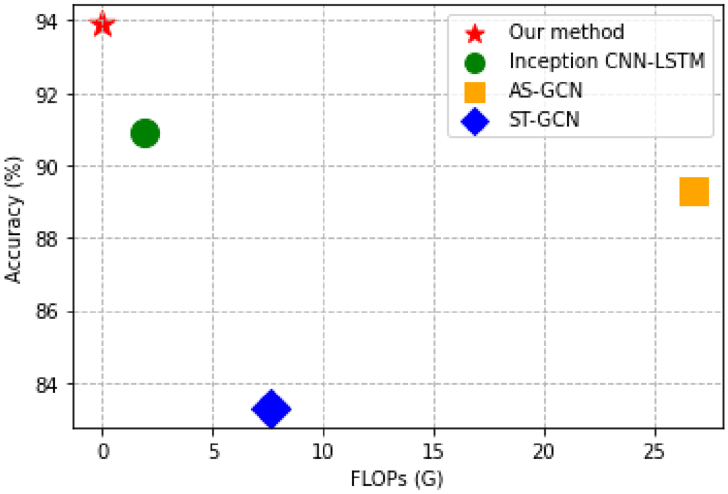


Fig. 2. FLOPs v.s. accuracy on the NTU RGB+D CS protocol.

rectional LSTM network that processes CTVs and TDVs to model temporal information, and a lightweight model that boosts computational speed.

The rest of this paper is organized as follows. Chapter 2 reviews some related works on action recognition. Chapter 3 presents our proposed method. Chapter 4 presents our experiment results and ablation study. Chapter 5 concludes this paper.

2 Related Work

Most of the earlier approaches design handcrafted features to represent a human body. In [33], skeletons are modeled by rotations and translations in Lie group, which are then classified by a combination of dynamic time warping, Fourier temporal pyramid representation and linear support vector machine (SVM). In [34], use the covariance matrices are used to encode skeleton joint locations over time as a discriminative descriptor. In [35], joint data can also be represented by the parameters of ranker by the rank pooling method. However, since these handcrafted features are dataset-dependent, they are not capable of capturing all the information at the same time, limiting their performance.

With the development of deep learning, such as recurrent neural network (RNN) and convolutional neural network (CNN), RNN-based and CNN-based methods have attracted more and more attention. RNN-based methods show promising performance due to their strengths to capture temporal dependencies in sequential data. In [36], human skeletons are divided into five body parts and then a hierarchical bidirectional RNN model is applied to recognize actions. Reference [31] proposes a part-aware LSTM model, which splits a LSTM cell into five sub-cells corresponding to these body parts, i.e. torso, two arms, and two legs. A tree-like structure for human body and a gating mechanism to handle the noise and occlusion in 3D skeletons are proposed in [5]. The work [37] introduces an end-to-end spatial and temporal attention model, which learns to selectively focus on discriminative joints and frames. CNN-based methods convert skeleton information into a pseudo image and apply a convolution neural network, such as ResNet[38], to extract features. Liu et al. [39] transform the view-invariant skeleton information into a series of color images, while[40] proposes to use CNN to learn long-term temporal information of skeleton sequences and then uses a Multi-Task Learning Network (MTLN) to incorporate spatial structural information. It is proposed in [41] to represent the joint coordinate sequences as an image by treating xyz coordinates as image channels. Reference [18] also encodes the relationship between joints by three matrices and applies an Inception CNN-LSTM network to extract spatial and temporal information.

Recently, GCN-based methods become popular for this task due to their expressive power for skeleton data. A skeleton sequence can be represented as a graph, where joints correspond to vertices and bones correspond to edges. These methods can model the kinematic structure of human bodies more naturally than RNN-based methods and CNN-based methods. Reference [11] first proposes to use GCNs for skeleton-based action recognition. A skeleton sequence encoded as a graph, which consists of intra-body edges between joints and inter-frame edges between consecutive frames, which are fed to ST-GCN to learn both spatial and temporal patterns. 2s-AGCN [29] is proposed to learn the optimal edges in the spatial graph and to exploit the second-order information. Skeleton data is represented as a directed acyclic graph (DAG) based on the kinematic dependency between the joints and bones in [30]. An action-structural graph is proposed in [12] to capture action-specific later dependencies and to represent higher order dependencies.

3 Proposed Method

The key to success in skeleton-based action recognition is to extract discriminative features from a sequence of skeleton data. In this work, we consider 3-dimensional joints coordinates from two persons in a sequence of video frames. Instead of directly taking the original coordinates as inputs, we introduce two data representations to model spatial interaction information and joint dynamics. In order to make this paper self-contained, we first review stacked bidirectional LSTM. Then, we propose our data representations. Finally, we introduce our complete network architecture, which is a stacked bidirectional LSTM network.

4 Overview of Stacked Bidirectional LSTM

Long Short-Term Memory (LSTM) [42] is a type of recurrent neural network capable of learning long-term dependencies. Fig. 3 depicts the structure of a LSTM neuron. The transition equations are formulated as follows:

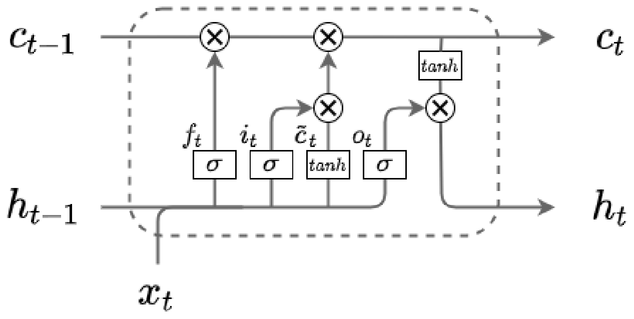
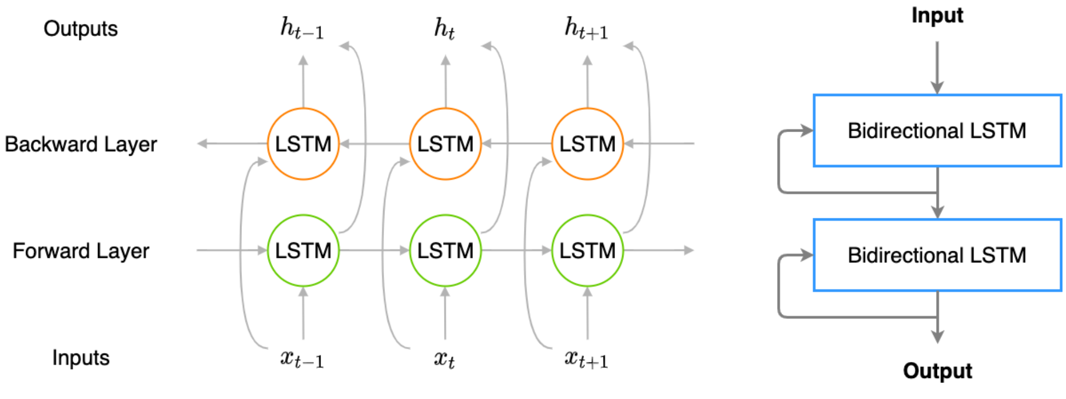


Fig. 3. The architecture of a common LSTM neuron.



(a) bi-LSTM (b) stacked bi-LSTM

Fig. 4. The architecture of a bidirectional LSTM network and a stacked bidirectional LSTM network.

$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i) \tag{1}$$

$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f) \tag{2}$$

$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o) \tag{3}$$

$$\tilde{c}_t = \sigma(W_c[x_t, h_{t-1}] + b_c) \tag{4}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \tag{5}$$

$$h_t = o_t \circ \sigma(c_t) \tag{6}$$

With the input x_t and the previous hidden state h_{t-1}, three control signals, i_t, f_t, and o_t are generated. The input gate signal i_t controls how much input information should be taken. The forget gate signal f_t decides which part of the current cell state will be remembered or forgot. The output gate signal o_t determines what information should be output. W and b are the trainable weights and bias for each gate signal. Given an input sequence $x = (x_0, ..., x_{T-1})$, we can derive a sequence of cell states $c = (c_0, ..., c_{T-1})$ and a sequence of hidden states $h = (h_0, ..., h_{T-1})$.

In order to utilize the forward and backward information, bidirectional LSTM (bi-LSTM) [43] contains two hidden layers of LSTM with opposite directions, where the first layer learns the input sequence and the second layer learns the reverse of the input sequence as shown in Fig. 4(a). For every point in bi-LSTM, the outputs are obtained based on the past and the future context information, which is beneficial to learn more comprehensive temporal dependencies. Take the action "exchanging objects" in SBU for example. At the moment of "exchanging", we can leverage the past action "giving objects" information and the future action "receiving objects" information, which are both important to learn the characteristic of the action.

A Stacked LSTM [44] is an extension of LSTM model that consists of multiple LSTM layers. In every layer of the stacked LSTM, we can create more complex features based on the outputs of the previous LSTM layer. Therefore, by stacking LSTM layers, the model can learn deeper and more accurate descriptions. In this work, we leverage a stacked bidirectional LSTM as shown in Fig. 4(b).

5 Data Representations

In order to extract discriminative features from a sequential skeleton data, we propose two data transformation techniques to represent spatial interaction information and joint dynamics. Let the two skeletons sequence of a frame at time t be a vector $v_t \in R^{2 \cdot J \times 3}$ represents the 3-dimensional coordinates at time t, where J is the number of joints per person. We convert v_t to Coordinate Transformation Vectors $CTV_t \in R^{2 \cdot J \times 4}$ and Temporal Difference Vector $TDV_t \in R^{2 \cdot J \times 4}$ as Fig. 5.

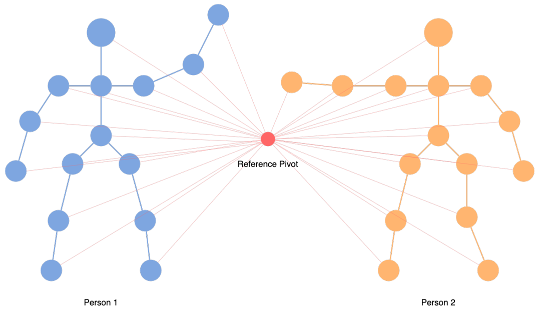


Fig. 5. Reference pivot and CTV.

5.1 Coordinate Transformation Vector

We first compute the reference *pivot*, which is defined as the center point of all $2 \cdot J \cdot T$ joints of two persons from all T frames. Next, we construct CTV_t for time t, which contains the relative coordinates of all joints to the pivot and their Euclidean distance to the pivot. Fig. 5 illustrates the concept.

$$pivot = \frac{1}{2 \cdot T \cdot J} \cdot \sum_{t=1}^{T} \sum_{j=1}^{J} v_t[1, j, 1:3] + v_t[2, j, 1:3] \tag{7}$$

$$CTV_t[i, j, 1:3] = v_t[i, j, 1:3] - pivot, \forall i = 1, ..., 2, j = 1, ..., J, t = 1, ..., T \tag{8}$$

$$CTV_t[i, j, 4] = \| v_t[i, j, 1:3] - pivot \|, \forall i = 1, ..., 2, j = 1, ..., J, t = 1, ..., T \tag{9}$$

The pivot is the mean joint throughout the time. There are two properties in this representation:

Spatial normalization. Coordinates are camera setup dependent. For example, in Fig. 6, the two persons appear at the upper-left and the lower-right corners of a picture. After transformation, the relative coordinates are restricted to the region centered at the pivot. This helps a model focus on the difference among actions instead of their absolute locations.

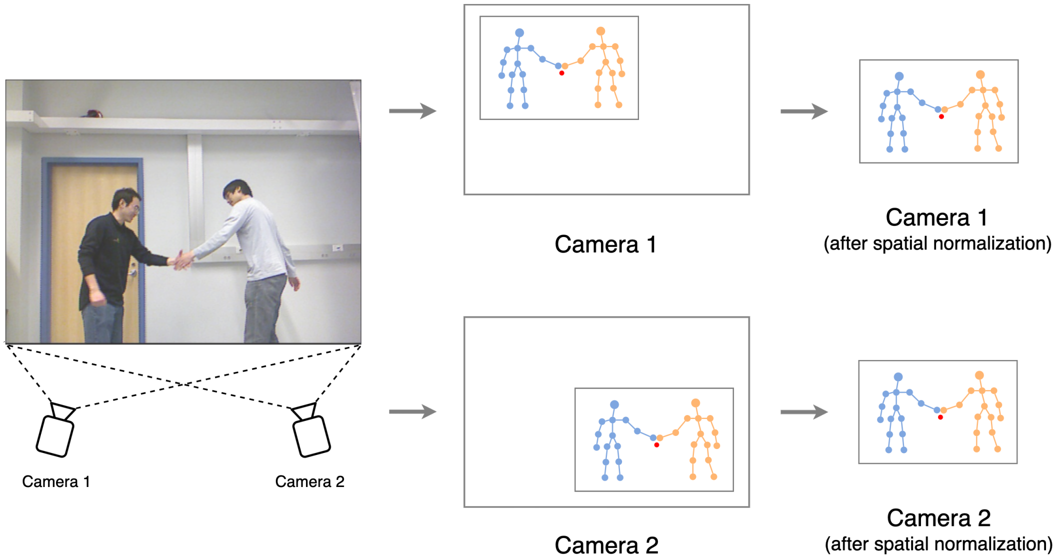


Fig. 6. An example of spatial normalization.

Spatial modeling of interaction. The interaction between two persons can be better represented through the pivot. Fig. 7 illustrates a "hand shaking" example. At the beginning, the two persons approach to the pivot. Then, they hold each other's hand, shake, and move away from each other. All these actions, when represented relative to the pivot, seem to be more informative.

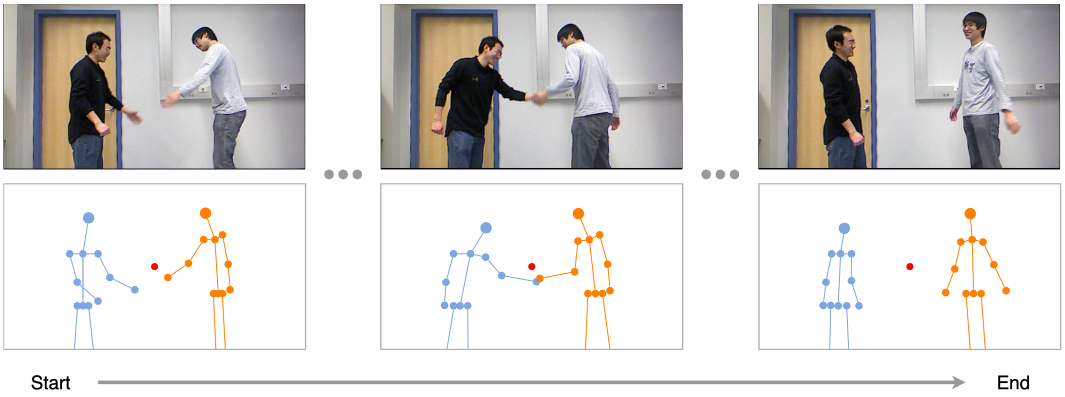


Fig. 7. An example of spatial modeling with respect to the pivot.

5.2 Temporal Difference Vector

We construct TDV_t to model joint dynamics. The purpose is to find the difference of each joint between frames, and focus more on their movements. The process can be formulated as follows:

$$TDV_t[i,j,1:3] = v_t[i,j,1:3] - v_{t-1}[i,j,1:3], \forall i = 1,...,2, j = 1,...,J, t = 2,...,T \tag{10}$$

$$TDV_t[i,j,4] = \parallel v_t[i,j,1:3] - v_{t-1}[i,j,1:3] \parallel, \forall i = 1,...,2, j = 1,...,J, t = 2,...,T \tag{11}$$

We calculate TDVs by finding the temporal difference in Eq. (10) and the Euclidean distance of each joint in Eq. (11) between adjacent frames. There are two properties in this representation:

Robustness to camera setup. This enables us to focus on the movements of joints, thus removing the impact of camera setup.

Joint dynamic modeling. We can directly model joint dynamics, instead of counting on the network to learn.

6 Network Architecture

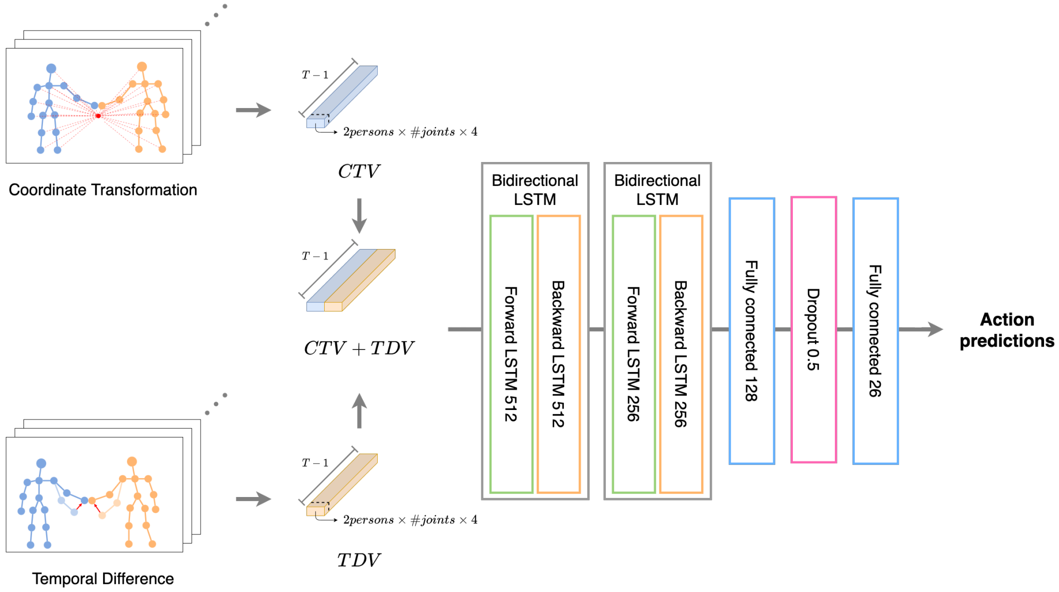


Fig. 8. The proposed network architecture.

Fig. 8 depicts the complete network architecture. First, we concatenate CTVs and TDVs as our inputs. Next, we leverage a 2-layer stacked bidirectional LSTM with 512 neurons and 256 neurons, respectively, to learn deep temporal forward and backward information. Finally, we apply a 2-layer fully connected network, which contains 128 and 26 neurons, with a dropout layer to predict the score of each action type.

7 Evaluation Results

7.1 Datasets

We adopt three datasets in this work. **SBU** [13] is a small-scale dataset of mutual interaction recognition captured by Microsoft Kinetic, providing 282 videos (around 2$\sim$3 seconds for each clip) with sequences of skeleton data. SBU dataset contains eight interactions (approaching, departing, pushing, kicking, punching, exchanging objects, hugging, and shaking hands) and seven participants. In each frame, skeleton information is presented as 3D coordinates over 15 joints per person. In this work, we follow the five-fold cross validation protocol suggested by the authors and report the average recognition accuracy by comparing different models.

NTU RGB+D [31] is the most widely used benchmark dataset for skeleton-based action recognition, containing 56,880 RGB videos with 3D skeleton data collected by Microsoft Kinetic V2. The action samples are performed by 40 volunteers and categorized into 60 action classes. Each skeleton data consists of 3D coordinates of 25 body joints from at most 2 persons per frame, captured by 3 cameras from different horizontal angles: -45°, 0°, and 45°. While this dataset is not specially designed for mutual interaction recognition, it contains 11 interaction classes (slapping, kicking, pushing, patting on the back, pointing finger, hugging, giving object, touching pocket, shaking hands, walking toward, and walking apart) with a total of 10,347 videos. The authors recommended two benchmarks: 1) cross-subject (CS): training data comes from 20 subjects, and validation data comes from the remaining 20 subjects. 2) cross-view (CV): training data comes from camera 2 and camera 3, and testing data comes from camera 1.

NTU RGB+D 120 [32] is the extended version of the NTU RGB+D dataset captured by Microsoft Kinetic V2, containing 114,480 videos performed by 106 volunteers with 60 additional classes. There are 15 more mutual action classes (hit with object, wield knife, knock over, grab stuff, shoot with gun, step on foot, high-five, cheers and drink, carry object, take a photo, follow, whisper, exchange things, support somebody, and rock-paper-scissors) with a total of 24,794 videos. Similarly, the authors suggested two benchmarks: 1) CS: training data comes from 53 subjects, and validation data comes from the remaining 53 subjects. 2) CV: training data comes from cameras with even IDs, and validation data comes from cameras with odd IDs.

7.2 Implementation Details

Data Preprocessing. Videos in the above datasets may differ in length. In order to align the length of input data, we use interpolation to make the number of frames consistent to 100. Furthermore, in many actions, there exists an active-passive relationship between two persons, such as “giving object” in SBU. In order to reduce the bias on the order of active and passive parties, we extend each dataset by swapping all frames by 180 degrees along the *x*-axis.

Training Procedure. Our model is trained using the Adam optimizer [45] with a dropout rate 0.5 and a batch size 64. At the beginning, we set the learning rate to 0.001 for 100 epochs. Then, we decrease the learning rate to 0.0005 for further optimization.

7.3 Ablation Studies

We perform experiments on NTU RGB+D for ablation studies and report the top-1 accuracy in Table 1. We design 6 different input features on the same stacked bidirectional LSTM model to observe the strengths of our proposed data representations.

Using Original Coordinates. Here, we directly input original data, which contains the 3D coordinates of every joint and their Euclidean distances to the origin. From Table 1 (a), we can observe a significant gap between the result of CS protocol and CV protocol. Since CV protocol splits training and validation data by different viewing angles, the results indicate that using the original coordinates without spatial normalization is vulnerable to different camera setups.

TDVs. Next, we validate the effectiveness of TDVs. By comparing (a) and (b) in Table 1, we can observe a notable improvement, especially in CV protocol. The results in CS protocol

Table 1: Ablation studies on NTU RGB+D.

	Input	Cross Subject	Cross View
(a)	Original coordinates	71.4%	10.9%
(b)	**TDVs only**	**79.1%**	**84.0%**
(c)	CTVs only with two separated reference pivots on each person	85.2%	89.3%
(d)	**CTVs only with a center reference pivot**	**93.3%**	**94.6%**
(e)	CTVs only with a non-fixed center pivot through time	92.7%	94.2%
(f)	**CTVs + TDVs**	**93.9%**	**95.6%**

demonstrate the ability of TDVs in modeling joint dynamics. The results in CV protocol further show that TDVs are much more robust to different camera setups.

Spatial Normalization of CTVs. We design another input feature vectors in (c) for ablation study. First, we define the center point of each person as the reference pivot, so there are two pivots. Next, we transform the original coordinates feature vectors for each person based on its own pivot. With these feature vectors, we can keep the spatial normalization characteristic of the original CTVs, but we cannot extract mutual-person interaction information. By comparing (a) and (d), we observe that spatial normalization can significantly boost the performance, particularly in CV protocol.

Spatial Interaction Information Modeling of CTVs. In Table 1 (d), we consolidate the reference pivots of each person to the same one, which is the same as that defined in Eq. (7). It allows us to collect more spatial interaction information rather than counting on the network only. From (c) and (d), we can see an obvious improvement in both of CS and CV protocol.

Stability of CTVs. Video frames are sequential data. In Table 1 (e), instead of finding a fixed reference pivot throughout all frames, we obtain a reference pivot per frame at the center of the two persons. Thus, reference pivots are dynamic. As shown in (d) and (e), using a fixed reference pivot is more preferable as it can better track the movement of the two persons. If we adopt per-frame pivots, we might obtain similar relative coordinates but lose the track of a person's moving features.

Proposed CTVs and TDVs. From Table 1 (a) to (e), we can observe that both CTVs and TDVs have their advantages. Table 1 (f) further the advantage of integrating CTVs and TDVs.

7.4 Comparisons with the State-of-the-arts

To verify the performance of our model, we perform experiments by considering top-1 accuracy on the SBU, NTU RGB+D, and NTU RGB+D 120 datasets. In order to verify the efficiency of our model, we further consider computational complexity on the NTU RGB+D dataset.

SBU. On the SBU dataset, we compare our method with 10 other approaches and report the results in Table 2. The experiments include handcrafted methods, RNN-based methods, and CNN based method. Our method shows the highest accuracy, surpassing the previous best approach, Inception CNN-LSTM [18], by 0.2%.

NTU RGB+D and NTU RGB+D 120. In order to verify the robustness of our model, we conduct experiments on large-scale datasets, NTU RGB+D and NTU RGB+D 120, and the results are

Table 2: Results on SBU.

Method	Top-1 Accuracy
Co-occurence LSTM [4]	90.4%
Deep LSTM (reported by [4])	86.0%
ST-LSTM [5]	93.3%
DM-3DCNN [6]	93.7%
VA-LSTM [7]	97.2%
Wu et al. [16]	91.0%
Two-stream GCA-LSTM [8]	94.9%
LSTM-IRN [17]	98.2%
Inception CNN-LSTM [18]	98.6%
Our method	**98.8%**

Table 3: Results on NTU RGB+D and NTU RGB+D 120.

Method	Cross Subject	Cross View
ST-LSTM [5]	83.0%	87.3%
GCA-LSTM [9]	85.9%	89.0%
Two-stream GCA-LSTM [8]	87.2%	89.9%
FSNET [10]	74.0%	80.5%
ST-GCN [11]	83.3%	87.1%
AS-GCN [12]	89.3%	93.0%
LSTM-IRN [17]	90.5%	93.5%
Inception CNN-LSTM [18]	90.9%	93.9%
Our method	**93.9%**	**95.6%**

(a) NTU RGB+D.

Method	Cross Subject	Cross View
ST-LSTM [5]	63.0%	66.6%
GCA-LSTM [9]	70.6%	73.7%
Two-stream GCA-LSTM [8]	73.0%	73.3%
FSNET [10]	61.2%	69.7%
ST-GCN [11]	78.9%	76.1%
AS-GCN [12]	82.9%	83.7%
LSTM-IRN [17]	77.7%	79.6%
Inception CNN-LSTM [18]	78.1%	80.4%
Our method	**83.9%**	**83.4%**

(b) NTU RGB+D 120.

reported in Table 3 (a) and Table 3 (b), respectively. Our model is compared with 8 other methods, including CNN-based method [10], RNN-based methods [5, 9, 8, 17, 18], and GCN-based [11, 12] methods. On NTU RGB+D, our model outperforms all previous methods, exceeding Inception CNN-LSTM, which is the second best, by 3.0% in the cross-subject protocol and by 1.7% in the cross-view protocol. On NTU RGB+D 120, which is more challenging for its larger dataset size and more action categories, our model can still outperform or be competitive to almost all previous works. Compared with AS-GCN [12], our model surpasses it by 1.0% in CS protocol and slightly loses by 0.3% in CV protocol.

Model Complexity. An important characteristic of our model is its low complexity. We demonstrate the lightweightness of our model on NTU RGB+D by comparing with 3 state-of-the-art methods, one RNN-based method [18] and two GCN-based methods [12, 5]. We evaluate complexity by the number of parameters and FLOPs, which impact both training and inference efficiency. Table 4 shows the comparison on two metrics. The ratios following parameter number and FLOPs

Table 4: Results of model complexity on NTU RGB+D.

Method	Cross Subject	Cross View	#Params	Ratio	FLOPs	Ratio
Our method	**93.9%**	**95.5%**	**6.43M**	**1x**	**0.000138G**	**1x**
Inception CNN-LSTM [18]	90.9%	93.9%	7.65M*	1.19x	1.90G*	13,768x
AS-GCN [12]	89.3%	93.0%	9.50M†	1.48x	26.76G†	193,913x
ST-GCN [5]	83.3%	87.1%	3.10M†	0.48x	16.32G†	118,261x

* The results are measured by implementing the released code.
† The results are obtained from [46].

reflects that our model is much more lightweight than the other models, especially those GCN-based ones, owing to its relatively simple architecture.

8 Conclusions

In this work, we propose a novel method for two-person mutual action recognition. First, we propose two data representations, Coordinate Transformation Vectors and Temporal Difference Vectors, for capturing spatial interaction information and joint dynamics. Then, we introduce a stacked bidirectional LSTM to learn the deep forward and backward temporal dependencies. Through experiments on the small-scale SBU dataset and the large-scale NTU RGB+D and NTU RGB+D 120 datasets, we demonstrate that our method outperforms the state-of-the-art approaches with higher accuracy and much less complexity. A possible future work is to extend the current model for real-time action recognition, which can validate the robustness of our model.

References

[1] Poppe R. A survey on vision-based human action recognition. Image and Vision Computing. 2010;28(6):976-90.

[2] Weinland D, Ronfard R, Boyer E. A survey of vision-based methods for action representation, segmentation and recognition. Computer Vision and Image Understanding. 2011;115(2):224-41.

[3] Aggarwal JK, Ryoo MS. Human Activity Analysis: A Review. ACM Comput Surv. 2011;43(3).

[4] Zhu W, Lan C, Xing J, Zeng W, Li Y, Shen L, et al. Co-Occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks. Proceedings of the AAAI Conference on Artificial Intelligence. 2016;30(1).

[5] Liu J, Shahroudy A, Xu D, Wang G. Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition. 2016:816-33.

[6] Hernandez Ruiz A, Porzi L, Rota Bulò S, Moreno-Noguer F. 3D CNNs on Distance Matrices for Human Action Recognition. 2017:1087–1095.

[7] Zhang P, Lan C, Xing J, Zeng W, Xue J, Zheng N. View Adaptive Recurrent Neural Networks for High Performance Human Action Recognition From Skeleton Data. 2017.

[8] Liu J, Wang G, Duan LY, Abdiyeva K, Kot AC. Skeleton-Based Human Action Recognition With Global Context-Aware Attention LSTM Networks. IEEE Transactions on Image Processing. 2018;27(4):1586-99.

[9] Liu J, Wang G, Hu P, Duan LY, Kot AC. Global Context-Aware Attention LSTM Networks for 3D Action Recognition. 2017.

[10] Liu J, Shahroudy A, Wang G, Duan LY, Kot AC. Skeleton-Based Online Action Prediction Using Scale Selection Network. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020;42(6):1453-67.

[11] Yan S, Xiong Y, Lin D. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. Arxiv:180107455. 2018.

[12] Li M, Chen S, Chen X, Zhang Y, Wang Y, Tian Q. Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition. 2019.

[13] Yun K, Honorio J, Chattopadhyay D, Berg TL, Samaras D. Two-person interaction detection using body-pose features and multiple instance learning. 2012:28-35.

[14] Ji Y, Ye G, Cheng H. Interactive body part contrast mining for human interaction recognition. 2014.

[15] Ji Y, Cheng H, Zheng Y, Li H. Learning contrastive feature distribution model for interaction recognition. Journal of Visual Communication and Image Representation. 2015;33:340-9.

[16] Wu H, Shao J, Xu X, Ji Y, Shen F, Shen HT. Recognition and Detection of Two-Person Interactive Actions Using Automatically Selected Skeleton Features. IEEE Transactions on Human-Machine Systems. 2018;48(3):304-10.

[17] Perez M, Liu J, Kot AC. Interaction Relational Network for Mutual Action Recognition. IEEE Transactions on Multimedia. 2021:1-1.

[18] Liang CJ. On Two-Person Mutual Action Recognition by Deep Learning. Taiwan: National Chiao Tung University; 2020.

[19] Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, et al. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. 2016:20-36.

[20] Feichtenhofer C, Pinz A, Zisserman A. Convolutional Two-Stream Network Fusion for Video Action Recognition. 2016.

[21] Carreira J, Zisserman A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. 2017.

[22] Tran D, Wang H, Torresani L, Ray J, LeCun Y, Paluri M. A Closer Look at Spatiotemporal Convolutions for Action Recognition. 2018.

[23] Keselman L, Iselin Woodfill J, Grunnet-Jepsen A, Bhowmik A. Intel RealSense Stereoscopic Depth Cameras. 2017.

[24] Zhang Z. Microsoft Kinect Sensor and Its Effect. IEEE MultiMedia. 2012;19(2):4-10.

[25] Toshev A, Szegedy C. DeepPose: Human Pose Estimation via Deep Neural Networks. 2014.

[26] Wei SE, Ramakrishna V, Kanade T, Sheikh Y. Convolutional Pose Machines. 2016.

[27] Cao Z, Simon T, Wei SE, Sheikh Y. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. 2017.

[28] Cheng B, Xiao B, Wang J, Shi H, Huang TS, Zhang L. HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation. 2020.

[29] Shi L, Zhang Y, Cheng J, Lu H. Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. 2019.

[30] Shi L, Zhang Y, Cheng J, Lu H. Skeleton-Based Action Recognition With Directed Graph Neural Networks. 2019.

[31] Shahroudy A, Liu J, Ng TT, Wang G. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. 2016.

[32] Liu J, Shahroudy A, Perez M, Wang G, Duan LY, Kot AC. NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020;42(10):2684-701.

[33] Vemulapalli R, Arrate F, Chellappa R. Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group. 2014:588-95.

[34] Vemulapalli R, Arrate F, Chellappa R. Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group. 2014:588-95.

[35] Fernando B, Gavves E, Oramas JM, Ghodrati A, Tuytelaars T. Modeling Video Evolution for Action Recognition. 2015.

[36] Du Y, Wang W, Wang L. Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition. 2015.

[37] Song S, Lan C, Xing J, Zeng W, Liu J. An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. Proc AAAI Conference on Artificial Intelligence. 2017;31(1).

[38] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. 2016.

[39] Liu M, Liu H, Chen C. Enhanced skeleton visualization for view invariant human action recognition. Pattern Recognition. 2017;68:346-62.

[40] Ke Q, Bennamoun M, An S, Sohel F, Boussaid F. A New Representation of Skeleton Sequences for 3D Action Recognition. 2017.

[41] Du Y, Fu Y, Wang L. Skeleton based action recognition with convolutional neural network. 2015:579-83.

[42] Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computation. 1997;9(8):1735-80.

[43] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks. 2005;18(5):602-10.

[44] Goldberg Y. A Primer on Neural Network Models for Natural Language Processing. Journal of Artificial Intelligence Research. 2016;57(1):345–420.

[45] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. Arxiv:14126980. 2017.

[46] Song YF, Zhang Z, Shan C, Wang L. Constructing Stronger and Faster Baselines for Skeleton-based Action Recognition. Arxiv:210615125. 2021.