Learning Latent Global Network for Skeleton-based Action Prediction

Qiuhong Ke, Mohammed Bennamoun, Hossein Rahmani, Senjian An, Ferdous Sohel, and Farid Boussaid

Abstract—**Human actions represented with 3D skeleton sequences are robust to clustered backgrounds and illumination changes. In this paper, we investigate skeleton-based action prediction, which aims to recognize an action from a partial skeleton sequence that contains incomplete action information. We propose a new Latent Global Network based on adversarial learning for action prediction. We demonstrate that the proposed network provides latent long-term global information that is complementary to the local action information of the partial sequences and helps improve action prediction. We show that action prediction can be improved by combining the latent global information with the local action information. We test the proposed method on three challenging skeleton datasets and report state-of-the-art performance.**

Index Terms—**Skeleton-based action prediction, adversarial learning, convolutional neural networks.**

I. Introduction

ACTION prediction aims to infer an action before the action is fully executed [1]. Action prediction is very important in a wide range of applications such as human-robot interaction, visual surveillance and health care systems [2].

Human actions generally occur in the 3D space. 3D skeleton sequences, i.e., 3D trajectories of human skeleton joints, provide more comprehensive information than RGB videos captured by 2D cameras [3]. Nowadays, accurate skeletons can be directly generated by depth sensors in real-time. Compared to RGB videos, the dimension of the skeleton sequences are much lower. Besides, skeleton sequences are more robust against clustered backgrounds and illumination changes. These advantages make skeleton sequences more attractive for action analysis [4], [5], [6], [7], [3], [8], [9]. In this work, we focus on action prediction based on 3D skeleton sequences.

To recognize actions from video sequences, the long-term global information of the complete action plays an important role[10], [11]. Reported works on action recognition focused on exploring the global information from the full video sequences using Hidden Markov Model (HMM) [12], [13] or models based on Conditional Random Field (CRF) [14]. Deep networks such as recurrent neural networks (RNNs) with Long-Short Term Memory (LSTM) neurons [15], [16] and Convolutional Neural Networks (CNNs) have also been used to learn the global representations from the full sequences for action recognition [17], [11], [18].

Compared to action recognition where the full sequences are available to learn the long-term global information, the testing sequences used in action prediction are partial sequences that contain an incomplete action execution [1]. For a complex human action containing a large variety of human postures and motions, the information provided by a partial sequence is usually different from the long-term global information of the full sequence. This makes action prediction from partial sequences more challenging than action recognition. Given that global action information is very important for action inference, we present a new method which aims to capture the latent global information of the partial sequences to improve action prediction. The main idea of the proposed method is to find a feature space, where the partial sequences are similar to the full sequences. During testing, the mapping of the partial sequence to the feature space generates a latent global representation that provides complementary global information, which helps predict actions more accurately.

More specifically, we minimize the difference between the full and partial sequences using adversarial leaning to learn the latent global information of the partial sequences. The adversarial leaning method is inspired by the Generative Adversarial Networks (GAN) [19], which were introduced to generate images that are similar to real images, and have been applied in different research areas such as object detection, anomaly detection and domain adaptation [20], [21], [22]. GAN simultaneously learns a discriminator (which evaluates whether an image is a fake image or a real one) and a generator (which aims to fool the discriminator such that the discriminator fails to discriminate between the fake and the real images). This paper focuses on inferring semantic action labels, rather than generating images. Instead of a generator, the proposed framework contains an inference network (I-Net in Fig. 1) and a discriminator (D-Net in Fig. 1). During training, the discriminator aims to evaluate whether the input is a full or a partial sequence. The inference network tries to retain a high accuracy of action inference and fool the discriminator so that it cannot distinguish the full and the partial sequences. Once the network is trained, given a testing partial sequence, the inference network is used to process

Qiuhong Ke is with School of Computing and Information Systems, The University of Melbourne,VIC Australia. E-mail: qiuhongke@gmail.com

Mohammed Bennamoun is with School of Computer Science and Software Engineering, The University of Western Australia, Crawley, Australia. E-mail: mohammed.bennamoun@uwa.edu.au

Hossein Rahmani is with School of Computing and Communications, Lancaster University, Lancashire, England. E-mail: h.rahmani@lancaster.ac.uk

Senjian An is with School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Bentley, Australia. E-mail: s.an@curtin.edu.au

Ferdous Sohel is with College of Science, Health, Engineering and Education, Murdoch University, Murdoch, Australia. E-mail: f.sohel@murdoch.edu.au

Farid Boussaid is with School of Electrical, Electronic and Computer Engineering, The University of Western Australia, Crawley, Australia. E-mail: farid.boussaid@uwa.edu.au

the testing sequence to infer the action class. This network is termed as a Latent Global Network (LGN), as it learns a hidden space where the full sequence that contains the full execution of an action shares the global information with the partial sequence.

Although the prediction performance for the partial sequences with small observation ratios is still inferior to the recognition performance of the full sequences, our experiments suggest that using the proposed LGN helps to improve the prediction performance and outperforms current state-of-the-art methods for action prediction and recognition.

The contributions of this paper are summarized as follows: **1)** We propose a new LGN based on adversarial feature learning to learn the latent global information of the partial sequences and improve action prediction. **2)** We demonstrate that exploiting both the latent global information and the local information improves the prediction accuracy. **3)** The proposed method achieves state-of-the-art performance for action prediction on three challenging skeleton datasets.

II. Related Works

Action prediction is relevant to action recognition. In this section, we review current works based on deep learning for both action recognition and action prediction.

Action Recognition: Action recognition based on RGB videos has been extensively explored using deep learning in recent years [23], [24], [25], [11], [17]. Most works have focused on learning the spatial information from each frame and the temporal dynamics from the video sequences using Convolutional Networks (ConvNet). Simonyan *et al.*[23] designed a two-stream ConvNet architecture to separately learn the complementary spatial and temporal information from videos for action recognition. The two-stream ConvNet contains a spatial stream that operates on individual video frames and a temporal stream which processes a stack of consecutive optical flow images. The probability scores of the two streams are fused for the final decision of the action class. Tran *et al.*[24] explored a 3D ConvNet to learn the spatio-temporal features of videos. The 3D ConvNet has shown to provide more capacity to model the temporal information compared to 2D ConvNet due to the spatio-temporal convolution and pooling operations, which result in a better performance. Wang *et al.*[11] proposed a temporal segment network to perform video-level predictions of actions. The temporal segment network models a sequence of snippets, which are randomly selected from different segments of the input video. The frame-level scores of the snippets are fused to generate the video-level probability scores. Varol *et al.*[25] designed an architecture with long-term temporal convolutions (LTC) to learn video-level representations. The temporal extent of representations is increased, while the spatial resolution is decreased, thus to keep the complexity of the proposed networks tractable. LSTM networks have also been used to model the temporal dependency in video sequences. Donahue *et al.*[17] developed a Long-term Recurrent Convolutional Network (LRCNs) for video recognition and description. LRCNs first use CNNs to extract features from each frame of the input video. The output features of the sequence are then fed to LSTMs to produce a sequence-level prediction.

Recognizing actions from RGB videos are challenging due to the complex backgrounds and illumination variations. Human actions can also be represented as trajectories of skeleton joints. Compared to RGB videos, skeleton sequences are more robust to variations in illumination and background variations. Previous works have extensively explored skeleton-based action recognition using recurrent neural networks [6], [3], [7], [8], [26], [27], [28], [29], [30], [31], [18]. Du *et al.*[6] designed a Hierarchical Recurrent Neural Network (HRNN) to learn hierarchically the features of skeleton sequences for action recognition. The first layer of HRNN consists of five bidirectional recurrent neural networks (BRNNs), which separately process five subsets of the entire skeleton. The subsequent layers hierarchically fuse the representations of the previous layers to generate the final representation for action recognition. Zhu *et al.*[3] proposed a regularized deep LSTM network, which contains a co-occurrence regularization to encourage the network to learn co-occurrence of joints for action recognition. Shahroudy *et al.*[7] developed a part-aware LSTM (P-LSTM), which contains part-based cells. Each cell of the part processes the joints of one body part. The outputs of all parts are combined as the final representation for action recognition. Liu *et al.*[8], [26] developed spatial-temporal LSTM to capture both the spatial and temporal information of skeleton sequences for action recognition. A trust gate is also introduced to handle noise in skeleton sequences. Liu *et al.*[27], [18] further introduced a global context-aware attention LSTM, which uses a global context memory cell to selectively focus on the informative joints for action recognition. CNNs have also been successfully used for skeleton-based action recognition [9]. Ke *et al.*[9] transformed each skeleton sequence into three clips. Each clip contains four frames, which depict the temporal information of the skeleton sequence. The clips are then processed with multiple CNNs, followed by a multi-task learning network for action recognition.

Action Prediction: Compared to action recognition, which aims to recognize actions from the full sequences of complete actions, the goal of action prediction is to predict an action from partial sequences of incomplete actions. Previous works have attempted to explore the evolution of features of partial sequences for action prediction based on RGB videos. Ryoo [32] designed an integral histogram of spatio-temporal features to capture the temporal evolution of feature distributions across the actions. Ryoo [32] also developed a dynamic bag-of-word to handle noisy observations. Kong *et al.*[33] designed a discriminative multi-scale model to capture the temporal structure of human actions using the features of partial videos and small segments. Lan *et al.*[2] designed a max-margin learning framework to predict actions based on a hierarchical feature representation, which contains HOG, HOF and MBH features to describe human appearance and motions from atomic to coarser levels. Besides hand-crafted features, deep networks have also been used for action prediction. Ke *et al.*[34] developed a temporal convolutional network which processes consecutive optical flow images to capture the temporal structure of an action. Ke *et al.*[35] designed a structural model based on LSTM networks to extract scene contextual information in two-person interactions. A ranking score fusion method is also introduced to combine different models to determine the final interaction class. Jain *et al.*[36] designed a weighted cross-entropy which prevents the network from over-fitting the partial sequences with small observation ratios. Aliakbarian *et al.*[37] proposed a new loss combining a weighted false positive cross-entropy with the standard cross-entropy to train the network for action prediction. Recently, Farha *et al.*[38] proposed a RNN and a CNN network for action anticipation. Unlike previous works that predict current action, in this paper, the goal is to anticipate future unseen actions from untrimmed videos. Ke *et al.*[39] introduced a time-conditioned network to achieve efficient and effective long-term anticipation. There are also some works focus on RGBD-based action prediction. Hu *et al.*[1] proposed a regression-based model to learn soft labels from partial sequences based on Local Accumulative Frame Feature (LAFF), which captures the historical information of each partial sequence. Ke [40] introduced a global regularizer and a Temporal-aware Cross-entropy to recognize actions from partial skeleton sequences. Liu *et al.*[41], [42] proposed a Scale Selection Network that contains a dilated convolutional network and a window scale selection scheme to extract the information of the current action from untrimmed videos for action prediction.

III. Latent Global Network

A complete human action generally comprises a large variety of human postures and motions. A partial action sequence without the global motion information is usually different from the associated full sequence, which contains the long-term global information of the action. Considering that the long-term global information is very important to understand the action class, we present a latent global network (LGN), which aims to improve action prediction by learning the latent global information of the partial sequence. The overall architecture of the proposed LGN is shown in Fig. 1. The main idea of the proposed LGN is to minimize the variation between the partial and full sequences using adversarial learning, thus to learn a feature space, where the full sequence shares the long-term global information with the partial sequence. In this section, we give detailed descriptions of the proposed method.

A. Network Architecture

As shown in Fig. 1, the proposed LGN contains an inference network (I-Net) and a discriminator (D-Net). During training, the inference network takes as inputs both a partial sequence and the associated full sequence. The hidden representations of the two inputs are separately fed to the discriminator, which aims to determine if the input is a partial or full sequence. The inference network is trained to jointly maximize the confusion of the discriminator and minimize the classification cross-entropy. In this work, we build the inference network with a deep residual network, whose feature encoding block is similar to ResNet-50 [43]. More specifically, the feature encoding block contains several convolutional layers which are the same as in the ResNet-50 [43], a global average layer, a dropout layer and a fully-connected layer for feature reduction. The output feature of the feature encoding block is fed to a fully connected layer and a Softmax layer to produce the probability scores of the action. The output feature of the feature encoding block is also fed as the input of the discriminator. The discriminator contains a fully connected layer and a Sigmoid layer.

More specifically, we denote the full and partial sequences as s_f and s_p, respectively. During training, s_f and s_p are separately fed to the inference network in parallel. We denote the hidden features of the full and partial sequences that are fed to the discriminator as h_f and h_p, respectively. The discriminator takes the hidden representations and classify the sequence as a full or partial sequence. The output of the discriminator (which is denoted as $D(\cdot)$) is a 1-dimensional probability score generated by a Sigmoid layer. The loss of the discriminator is the cross-entropy between the predicted probability and the ground-truth label. To train the discriminator, we set the ground-truth labels of all the full sequences to 1, and the ground-truth labels of all the partial sequences to 0. The loss of the discriminator is then formulated as:

$$\ell_D = -E[\log(D(h_f))] - E[\log(1 - D(h_p))] \tag{1}$$

The discriminator aims to determine whether the input is a full or a partial sequence, while the inference network aims to update the parameters to fool the discriminator such that the discriminator cannot classify the sequences correctly. The adversarial loss for the inference network is formulated as:

$$\ell_{Ia} = -E[\log(D(h_p))] \tag{2}$$

Equation (2) is inspired by the non-saturating game [44]. It encourages the inference network to learn a feature space, where the discriminator has difficulty distinguishing whether the input is a full or a partial sequence. Consequently, the full sequences share information with the partial sequences in learning action representations.

The output of the inference network (which is denoted as $I(\cdot)$) is an m-dimensional probability score generated by a Softmax layer, where m denotes the number of action classes. We denote the feature vector that is fed to the Softmax layer by $\mathbf{z} = [z_1, \cdots, z_m]^T \in \mathbb{R}^m$, which is used to produce the probability score of each class. The predicted probability of the i^{th} class is formulated as:

$$p_i = \frac{\exp(z_i)}{\sum\limits_{j=1}^{m} \exp(z_j)} \tag{3}$$

That is, $I(\cdot) = [p_1, \cdots, p_m]^T \in \mathbb{R}^m$. During training, besides minimizing the adversarial loss ℓ_{Ia}, the inference network also needs to minimize the multi-class cross-entropy of action inference, which is formulated as:

$$\ell_{Ic} = -E[y_{cf} \log I(s_f)] - E[y_{cp} \log I(s_p)] \tag{4}$$

where $I(s_f)$ and $I(s_p)$ are the outputs of the inference network for the full sequence and the partial sequence, respectively. $y_{cf} \log I(s_f)$ and $y_{cp} \log I(s_p)$ denote the losses of an individual sample. $E(\cdot)$ represents the mean operator of the losses

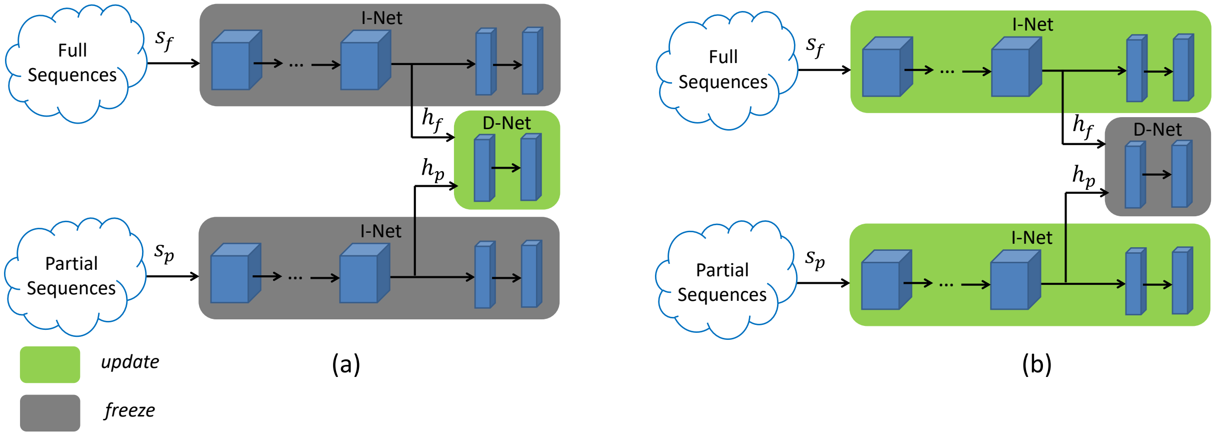


Fig. 1: Overall architecture of the proposed LGN, which is trained iteratively with two inner steps (a) and (b). 'I-Net', and 'D-Net' are abbreviations of the inference network and the discriminator, respectively. s_p and s_f denote a partial sequence and its associated full sequence. h_p and h_f denote the hidden feature representations of the two inputs, respectively.

of all the training samples. $I(s_f)$ and $I(s_p)$ denote the outputs of the inference network I-Net given the full sequence s_f and the partial sequence s_p, respectively. $y_{cf} \in \mathbb{R}^m$ is a one-hot vector. The non-zero element corresponds to the ground-truth action label of the input full sequence s_f, *e.g.*, if the action class of the sequence is q, the q^{th} element of y_{cf} equals to one while the other elements of y_{cf} equal to zero. Considering that the partial sequence contains less action information than the full sequence, we design a pseudo label $y_{cp} \in \mathbb{R}^m$ for the partial sequence. The ground-truth probability of the pseudo label y_{cp} is smaller than that of the full sequence y_{cp}, which equals to 1. We observe that using the pseudo label guarantees the convergence of the network. More specifically, if the action class of the partial sequence is q, the q^{th} element of y_{cp} equals to r while the other elements equal to zero. r is the observation ratio of the partial sequence, which is formulated as:

$$r = \frac{t}{T} \tag{5}$$

where t and T denote the numbers of frames of the partial and associated full sequences, respectively.

The loss of the inference network is the combination of the multi-class cross-entropy of action inference and the adversarial loss. More specifically,

$$\ell_I = \ell_{Ic} + \lambda \ell_{Ia} \tag{6}$$

where λ is the weight that balances these two terms.

B. Network Training

The D-Net is optimized using the loss function formulated in Equation (1), which is a binary classification loss that encourages the D-Net to classify the features of the full sequences as positive samples and the features of the partial sequences as negative samples. The I-Net is optimized with the loss function formulated in Equation (6), which contains a multi-class cross-entropy and an adversarial loss. The adversarial loss punishes partial sequences so that the output of the D-Net for the partial sequences is 1, which is different from the goal of the D-Net (i.e., classifies the partial sequences as negative samples). In this case, the parameters of the D-Net have to be fixed during the training of the I-Net, so that the D-Net is not affected by the I-Net and does not lose its ability to distinguish between full and partial sequences. On the other hand, the D-Net does not consider the difference between different actions, i.e., the partial sequences of all actions are treated as negative samples while the full sequences of all actions are treated as positive samples. In this case, the I-Net should also be fixed during the training of D-Net, so that it is not affected by the D-Net and does not lose its distinguishing ability between different actions. Therefore, the proposed LGN is trained iteratively in two inner steps: **1)** freeze the parameters of the inference network and train the discriminator, and **2)** freeze the parameters of the discriminator and train the inference network to confuse the discriminator and retain high accuracies of action inference. The overall algorithm is illustrated in Fig. 1 and summarised in Algorithm 1.

Algorithm 1 Training algorithm.

```
Initial the Inference network I-Net;
Freeze the parameters of I-Net and train the discriminator
  D-Net by minimizing the cost in Equation (1);
while not convergent do
    Freeze D-Net and update I-Net by minimizing Equation (6);
    Freeze I-Net and Update D-Net by minimizing the cost in Equation (1);
end
```

C. Network Input

The inputs of the proposed LGN include the full and partial sequences. The full sequences are provided by each dataset.

The partial sequences are segmented from each full sequence. Each partial sequence starts from the first frame of the full sequence, thus to retain the accumulative history information. More specially, we denote a full skeleton sequence as $s(1:J,1:T)$. J denotes the number of joints and T denotes the number of frames. We assume that the total number of all the partial and full sequences is n. The i^{th} partial sequence is then denoted as $s(1:J,1:\left[\frac{T\cdot i}{n}\right])$. The observation ratio of this partial sequence is $\frac{i}{n}$. Inspired by [9], we first transform each skeleton sequence into a color image. To be more specific, given a sequence with frame number T, we first arrange the sequence into a 2D array with size $T\times J\times 3$, where J is the number of joints in each frame of the sequence. Each channel of the 2D array is transformed into a gray image with a linear transformation. The transformation can be formulated as $T_i=\frac{S_i-\min(S_i)}{\max(S_i)-\min(S_i)},(i=1,2,3)$. S_i is the i^{th} channel of the array. $\min(S_i)$ and $\max(S_i)$ are the minimum value and the maximum value of S_i, respectively. The three generated gray images $T_i,(i=1,2,3)$ are combined as a color image. The three channels correspond to the three channels of the 3D coordinates of the skeleton joints. Each image is resized to 224×224 by performing bicubic interpolation to the image. Figure 2 shows some examples of generating images from skeleton sequences. Each row contains sequences of the same action performed by two different persons. It can be seen that the generated images of the same action contain a similar pattern.

IV. Join Latent Global and Local Information for Action Prediction

The proposed LGN aims to investigate latent global information for action inference. We also propose a local network, which aims to capture the local information of each sequence. An action usually contains variations in human postures and motions. Partial sequences of different observation ratios of the action generally contain discriminative local information. For example, a partial sequence with a small observation ratio contains some prior sub-actions, while a partial sequence with a larger observation ratio captures further progress of the action. The goal of the local network is to capture the local information of each sequence. In the experiment we have shown that the performance of using only LGN (i.e., without the local network) is higher than previous methods. The additional local network is to introduced demonstrate that exploiting both the latent global information and the local information improves prediction performance.

The architecture of the local network is the same as the inference network. More specifically, it contains five convolutional blocks (which are the same as ResNet-50 [43]), followed by an average pooling layer and two fully connected layers to generate action scores. The local network contains one input. We use all the skeleton sequences, including all the full and the partial sequences to train the local network. The parameters of the local network are updated by minimizing the classification cross-entropy between the output probabilities and the ground-truth action labels. During training, the local network and the global network are trained separately with their respective loss. More specifically, the local network is trained with the classification cross-entropy as formulated in Equation (4). The global network is trained iteratively using the training algorithm described in Algorithm 1. During inference, given a testing sequence, the inference networks of the global and the local networks are separately used to process the sequence and generate action scores. The scores of the global and local networks are averaged for the final decision of the action class. More specifically, given a testing sequence s_i, the output classification score of this sequence p_i is formulated as follows:

$$p_i=\frac{1}{2}(I_g(s_i)+I_l(s_i)) \tag{7}$$

where I_g and I_l denote the inference networks of the global and the local networks. $I_g(s_i)$ and $I_l(s_i)$ denote the output class probabilities of the two networks.

V. Experiments

The proposed method was evaluated on three skeleton datasets, i.e., NTU Dataset [7], SYSU 3D Human-Object Interaction (3DHOI) Dataset [45] and CMU Dataset[46]. In this section, we present the details of our experimental results.

A. Datasets

NTU Dataset [7] is currently the largest skeleton-based action dataset. It contains more than 4 million frames and 56000 sequences. Each skeleton contains 25 joints. This dataset was captured by three cameras placed at different locations and view points (80 view points in total). There are 60 action classes performed by 40 subjects. The actions consist of complex two-person interactions such as handshaking and one-person actions such as drinking. Due to the large intra-class diversity and the variation of view points, this dataset is very challenging.

SYSU 3DHOI Dataset [45] contains 480 sequences of 12 action classes, including playing with a cell phone, calling with a cell phone, pouring, drinking, moving a chair, sitting on a chair, packing a backpack, wearing a backpack, sweeping, mopping, taking something out from the wallet and taking out a wallet. This dataset is very challenging for action prediction due to the fact that some actions contain similar motions or the same operating object at the early temporal stage.

CMU Dataset [46] contains about 1 million frames and 2235 sequences categorized into 45 action classes [3] including sitting and running. All actions are performed by one person. Each skeleton contains 31 joints. This dataset is very challenging due to the large variation in the numbers of sequences.

For all datasets, we set the parameter λ in Equation (6) to 0.01. The parameter is selected by cross-validation. The number of units of the fully-connected layer in the feature encoding block of the inference network is set to 512. The numbers of units of the fully-connected layers in the inference network and the discriminator are set to the number of action classes and one, respectively. For each full sequence, we segmented 9 partial sequences with an increasing observation ratio from 0.1 to 0.9, with a step of 0.1. Each partial sequence starts from the first frame of the full sequence.

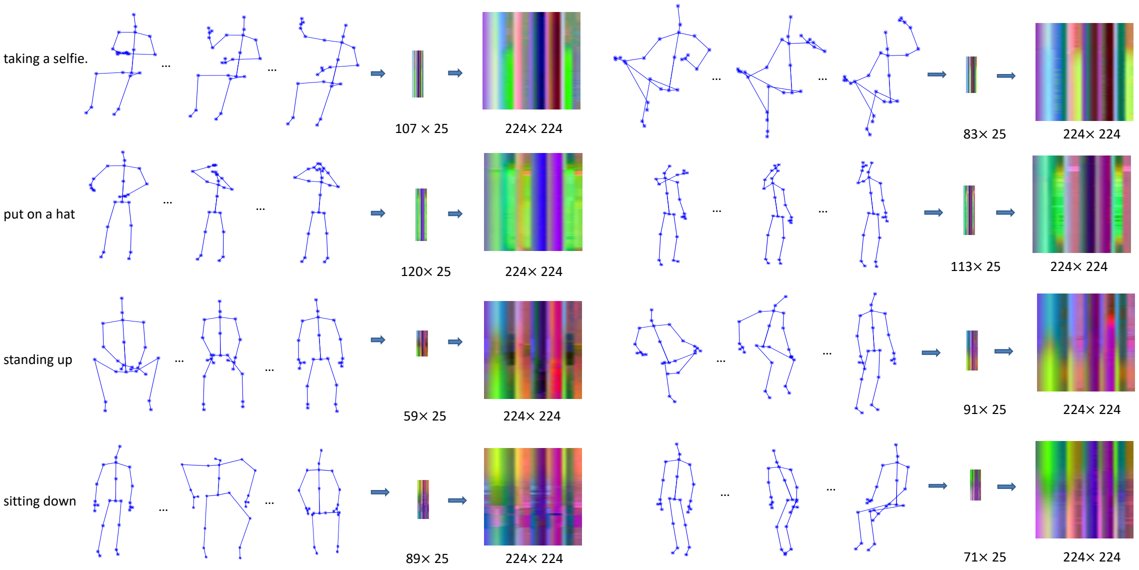


Fig. 2: Demonstration of image generation from skeleton sequences. Each row contains two sequences of the same action performed by two persons.

TABLE I: Action prediction performance comparison on the NTU dataset. Refer to Fig. 3 for more results.

Methods	Observation Ratio				
	0.2	0.4	0.6	0.8	1.0
[9]	8.34%	26.97%	56.78%	75.13%	80.43%
[36]	7.07%	18.98%	44.55%	63.84%	71.09%
[37]	27.41%	59.26%	72.43%	78.10%	79.09%
Local	29.32%	58.68%	70.81%	75.91%	76.47%
LGN	30.04%	61.78%	76.14%	81.57%	82.64%
Local+LGN	**32.12%**	**63.82%**	**77.02%**	**82.45%**	**83.19%**

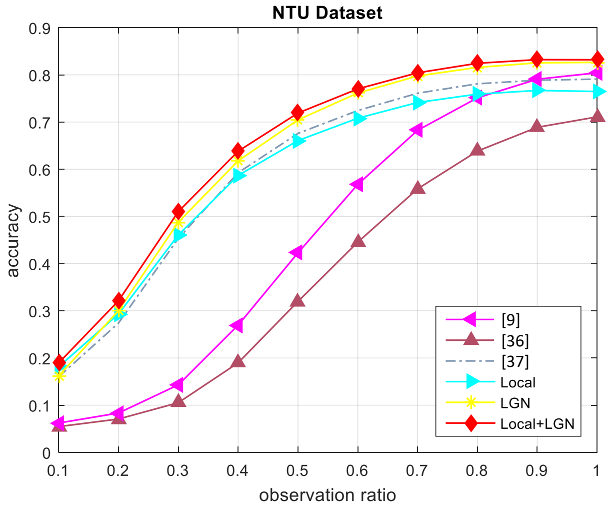


Fig. 3: Action prediction comparison on the NTU Dataset. The partial sequence with observation ratio r starts from the first frame to the $(rT)^{th}$ frame (T denotes the number of frames in the full sequence). (Best viewed in color)

The contributions of this work include **1)** a new **LGN** designed for action prediction, and **2)** combining the local network (**Local**) and the LGN (**Local+LGN**) for action prediction. The local network can be treated as an action recognition method except that the input consists of all the partial and full sequences. For each dataset, we first compare the LGN with the local network to show the benefit of learning the latent global information for action prediction. We further compare the proposed LGN and the Local+LGN to other action recognition and prediction methods [9], [36], [37], [1] to show the contributions of this work.

B. Results on the NTU Dataset

We followed the cross-subject testing protocol to evaluate this dataset. More specifically, the training set consists of the sequences of 20 subjects. The sequences of the other 20 subjects are used for testing. The results are shown in Table I and Fig. 3. It can be seen that the proposed LGN significantly outperforms the local network, especially at the late temporal stage. When the observation ratio is 0.4, i.e., using the first 40% of all frames of each full sequence to predict the action, the LGN achieves an accuracy of 61.78%. The prediction accuracy of the local network is 58.68%. The LGN outperforms the local network by 3.1%. The improvement of the LGN compared to the local network is more significant when the observation ratio increases, *e.g.*, the performance of the LGN with an observation ratio 0.6 is 76.14%, which is 5.33% better than the local network (70.81%). Compared to the local network, the LGN leverages adversarial learning to learn the latent global information, which provides a crucial cue to accurately infer actions. The improvements of the LGN clearly demonstrate the benefit of learning the global

TABLE II: Action recognition performance comparison on the NTU dataset.

Methods	Accuracy
Lie Group [5]	50.1%
Dynamic Skeletons [45]	60.2%
Hierarchical RNN [6]	59.1%
Deep RNN [7]	59.3%
Deep LSTM [7]	60.7%
Part-aware LSTM [7]	62.9%
ST-LSTM + Trust Gate [8]	69.2%
GCA-LSTM [27]	74.4%
GCA-LSTM (stepwise training) [18]	76.1%
SkeletonNet [47]	75.94%
Clips+CNN+MTLN [9]	79.57%
RotClips+MTCNN [48]	81.09%
Local	76.47%
LGN	82.64%
Local+LGN	**83.19%**

TABLE III: Action prediction performance comparison on the SYSU 3DHOI dataset. Refer to Fig. 4 for more results.

Methods	Observation Ratio				
	0.2	0.4	0.6	0.8	1.0
[1]	29.58%	35.42%	53.33%	58.75%	54.17%
[9]	26.76%	52.86%	72.32%	79.40%	80.71%
[36]	31.61%	53.37%	68.71%	73.96%	75.53%
[37]	56.11%	71.01%	78.39%	80.31%	78.50%
Local	57.15%	71.53%	78.69%	80.40%	77.78%
LGN	56.18%	72.60%	81.35%	84.28%	**83.33%**
Local+LGN	**58.81%**	**74.21%**	**82.18%**	**84.42%**	83.14%

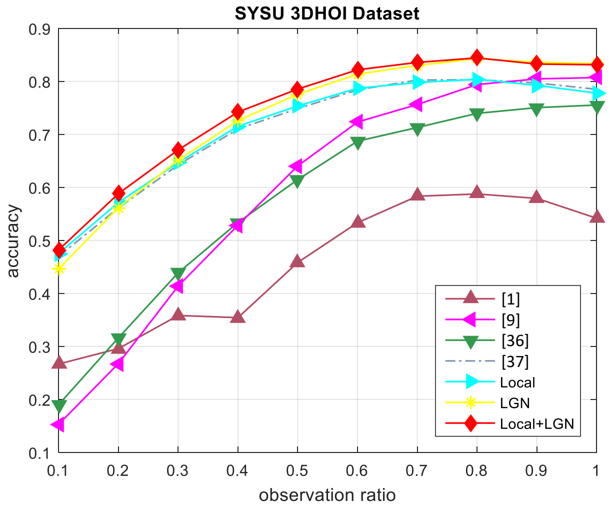


Fig. 4: Action prediction performance comparison on SYSU 3DHOI Dataset. (Best viewed in color)

information for action prediction. The proposed Local+LGN further improves the performance of the LGN. When the observation ratio is 0.4, the performance of the Local+LGN is 63.82%. Compared to the LGN (61.78%), the improvement of the Local+LGN is 2.04%. The local network aims to explore the local discriminative information, which is complementary to the latent global information learned by the LGN. The combination of the local network and the LGN thus results in a stronger action predictor and produces a better performance for action prediction.

We also compared the proposed method to recent state-of-the-art action recognition and prediction methods [9], [36], [37]. The work in [9] proposes a recognition network that trains only on full sequences. The network is optimized using the standard multi-class cross-entropy loss. The method in [36] trains on all partial and full sequences using a weighted loss function that applies different weights to the sequences. The work in [37] also trains all the partial and full sequences using a new loss that combines a weighted false positive cross-entropy and the standard cross-entropy. The above methods do not consider the missing global information in the partial sequences. In this paper, we introduce a LGN to exploit the missing global information, which produces a better understanding of the actions. The results are shown in Table I and Fig. 3. The recognition method [9] does not see the partial sequences during training, thus resulting in a worse prediction performance on the partial sequences during testing. When the observation ratio is 0.2, the performance of [9] and the proposed method is 8.34% and 32.12%, respectively. Compared to [9], the improvement of the proposed method is 23.78%. The prediction performance of [9] increases when the observation ratios of the partial sequences are close to 1 (i.e., testing on the full sequences). Similar to [9], [36] performs well only when the observation ratios of the testing sequences are close to 1. The prediction accuracy of [36] on observation ratio 0.2 is 7.07%, which is 25.05% worse than the proposed method. In [36], for each partial sequence, the weight of the weighted loss decreases exponentially with the difference of the number of frames between the partial sequence and the associated full sequence. This results in a worse performance at the early temporal stage. When the observation ratio is 0.6, [37] achieves an accuracy of 72.43%, which is 4.59% worse than the proposed method (77.02%). Compared to the proposed network, this method does not consider the global information for action prediction.

The proposed method also achieves state-of-the-art performance for action recognition. The comparisons of the proposed method with other action recognition methods are shown in Table II. Compared to the recent RotClips+MTCNN method [48], the improvement of the proposed method is 2.1% (from 81.09% to 83.19%).

C. Results on the SYSU 3DHOI Dataset

This dataset has been used for action prediction using the cross-subject setting [1]. In this setting, there are 30 training/testing splits, which are provided by [45]. In each split, the sequences of 20 subjects are used for training,

TABLE IV: Action recognition comparison on the SYSU 3DHOI dataset.

Methods	Accuracy
LAFF (SKL)[1]	54.2%
Dynamic Skeletons [45]	72.5%
ST-LSTM (Joint Chain) [26]	72.1%
ST-LSTM (Joint Chain) + Trust Gate [26]	74.8%
ST-LSTM (Tree) [26]	73.4%
ST-LSTM (Tree) + Trust Gate [26]	76.5%
Local	77.78%
LGN	**83.33%**
Local+LGN	83.14%

TABLE V: Action prediction comparison on the CMU dataset. Refer to Fig. 5 for more results.

Methods	Observation Ratio				
	0.2	0.4	0.6	0.8	1.0
[9]	33.72%	56.52%	71.33%	78.83%	80.64%
[36]	23.25%	46.99%	63.50%	71.48%	75.34%
[37]	71.08%	77.84%	81.58%	82.00%	81.07%
Local	72.62%	78.62%	81.53%	82.35%	81.62%
LGN	72.98%	79.73%	**83.86%**	84.65%	84.12%
Local+LGN	**74.79%**	**80.73%**	83.62%	**84.94%**	**84.53%**

and the sequences of the other 20 subjects are used for testing. Following [1], we report the results by averaging the accuracies over the 30 training/testing splits. The results are shown in Table III and Fig. 4. The proposed LGN outperforms the local network in 8 out of 10 cases. When the observation ratio is 0.6, the performance of the LGN is 81.35%. Compared to the local network (78.69%), the improvement of the proposed LGN is 2.66%. The Local+LGN further improves the performance of the LGN in 9 out of 10 cases. When the observation ratio is 0.4, the prediction performance of the Local+LGN is 72.60%, which is 1.61% better than the LGN (74.21%). The improvement of Local+LGN compared to the local network in this case is about 2.68% (from 71.53% to 74.21%). This clearly shows the benefits of combining the complementary strength of the local and global information for action prediction.

The proposed LGN and Local+LGN are also compared with state-of-the-art action prediction and recognition methods [1], [37], [9]. The results are shown in Table 4 and Fig. III. Similar to the results in the NTU dataset, [9] performs well only when the observation ratio is close to 1. It clearly shows the importance of training partial sequences for action prediction. The performance of [37] is 78.39% when the observation ratio is 0.6. The proposed LGN and Local+LGN outperform [37] by 2.96% and 3.79%, respectively. These improvements clearly show the advantages of learning the latent global information for action prediction.

We also show the action recognition comparison of the proposed LGN and Local+LGN with other methods. The results are shown in Table IV. Compared to the state-of-the-art recognition method [26], the proposed LGN improves the recognition performance by 6.83% (from 76.5 to 83.33%).

From Table 4 and Fig. IV, we can also see that the accuracies of most methods (including [37], the local network, the LGN, and the Local+LGN) on an observation ratio 0.8 are better than the accuracies on observation ratio 1. For example, the proposed method achieves an accuracy of 84.42% on observation ratio 0.8, which is 1.28% better than the accuracy on observation ratio 1 (83.14%). On this dataset, most actors at the last stage do not have any motions, which makes the actions ambiguous. In this case, the full sequences contain more noisy information compared to the partial sequences without the last temporal stage. This results in a worse performance on observation ratio 1.

D. Results on the CMU Dataset

For evaluation on this dataset, we followed the testing protocol of four-folder cross-validation introduced by [3]. The action prediction comparison of the proposed LGN and Local+LGN compared to other methods are shown in Table V and Fig. 5. The proposed Local+LGN improves the performance of the proposed LGN in 9 out of 10 cases and outperforms recent action recognition and prediction methods [9], [36], [37] on all observation ratios.

From Table V and Fig. 5, we can also see that the performance of the proposed method on observation ratio 1 (84.53%) is 9.74% better than the performance on observation ratio 0.2 (74.79%). The improvement between the two observation ratios on this dataset is smaller than that on the NTU dataset (51.07%) and on the SYSU dataset (24.33%). This is due to the fact that most actions on this dataset such as running are repetitive and periodic. Partial sequences with small observation ratios contain similar action information to the sequences with larger observation ratios, thus the performance at the early stage has a smaller gap compared to the performance at the late stage.

E. Analysis of the Weight of the Adversarial Loss

In this work, the weight parameter of the adversarial loss is selected by cross-validation. During cross-validation, we split the training dataset (performed by 20 subjects) into 70%/30% training/validation split, i.e., the sequences of 14 subjects (70%) out of the 20 subjects are used for training and the remaining 6 subjects (30%) are used as validation data. The validation performance of the proposed LGN with different weights of the adversarial loss are shown in Table VI. It can be seen that setting the weight to 0.01 produces the best validation result, and applying a larger weight to the adversarial loss results in a degradation of the action prediction performance. This is because the I-Net is optimized with a combination of the classification cross-entropy and the adversarial loss. The classification cross-entropy encourages the I-Net to distinguish between sequences of different actions. The adversarial loss forces the I-Net to fool the D-Net so that the output of the D-Net for all the partial sequences is the same as its output for the full sequences. In this case, the difference between all

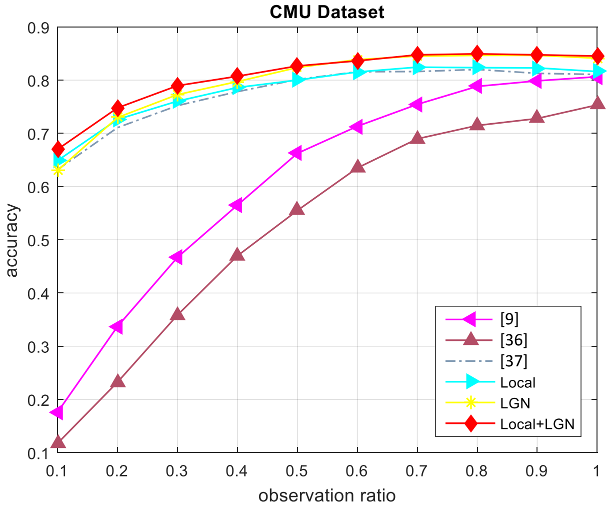


Fig. 5: Action prediction comparison on the CMU Dataset. (Best viewed in color)

the action classes is neglected. If the weight of the adversarial loss is too large, the I-Net will lose its ability to distinguish between different actions. As a result, the performance of action prediction will degrade.

TABLE VI: Validation Performance of action prediction using different weights of the adversarial loss on the NTU dataset.

Weight	Observation Ratio				
	0.2	0.4	0.6	0.8	1.0
0.005	27.17 %	54.49%	66.84%	72.44%	73.97%
0.01	**27.96%**	**55.26%**	**68.27%**	**74.14%**	**75.22%**
0.05	26.27 %	53.88%	66.95%	72.70%	74.16%
0.1	25.54 %	51.28%	65.39%	73.30%	74.83%
1	21.99%	49.72%	64.33%	71.07%	72.92%

F. Training the Adversarial Loss on the Full Sequences

In the proposed method, the adversarial loss penalises the partial sequences to ensure that the output of the discriminator for the partial sequences is 1. We have conducted an experiment with the adversarial loss penalising both the full and the partial sequences. The results are shown in Table VII. It can be seen that the results are similar. This is because the discriminator tries to classify the full sequences as positive samples, and the output of the discriminator for the full sequences is already forced to be 1 using the loss of the discriminator. In this case, an additional penalty of the full sequences (i.e., force the output of the discriminator for the full sequences to be 1) does not affect the convergence and the results are similar.

G. Visualizations of Action Prediction

We show predictions of our LGN method at different observation ratios of each sequence in Figure 6. For example, observation ratio 0.2 denotes that only the first 20% of the frames of a sequence are used to predict the action. The ground-truth action class of each sequence is shown on the left of the sequence. Incorrect predictions are shown in red and correct predictions in green. The first three rows show examples of correct predictions. These actions contain distinguishing motions and postures and are easy to predict. The last three rows show examples of failure. Some pairs of actions (e.g., cross hands in front and rub two hands together) are similar at the early temporal stage, which makes it difficult to distinguish between them when using a small observation ratio of the sequence. Some pairs of actions (e.g., typing on a keyboard and playing with a phone) contain the same motion and posture but involve different objects. It is difficult to predict the action class with only the skeleton information. In this case, the prediction is incorrect even when the full action sequence (i.e., observation ratio is 1) is used.

TABLE VII: Action prediction using different adversarial training methods on the NTU dataset.

Method	Observation Ratio				
	0.2	0.4	0.6	0.8	1.0
partial	**30.04%**	**61.78%**	**76.14%**	**81.57%**	82.64%
full & partial	29.42%	61.31%	75.24%	81.42%	**82.93%**

VI. Conclusion

In this paper, we presented a new Latent Global Network (LGN) for action prediction. The proposed LGN attempts to learn the latent global information, which is important to understand an action class. During training, the LGN leverages adversarial learning to encourage the partial sequences to be similar to the full sequences in a feature space. During testing, the partial sequences are mapped to the feature space for action inference. The proposed LGN has been shown to improve action prediction. We further demonstrated that the local information learned by the local network is complementary to the latent global information learned by the LGN and the combination of the LGN and the local network further improves the performance. The proposed LGN and Local+LGN achieve state-of-the-art performance for action prediction and recognition on challenging skeleton datasets.

Acknowledgments

We greatly acknowledge NVIDIA for providing a Titan XP GPU for the experiments involved in this research. This work was supported by the Australian Research Council under Grant DP150100294, Grant DP150104251, and Grant DE120102960. This paper used the NTU RGB+D Action Recognition Dataset made available by the ROSE Lab at the Nanyang Technological University, Singapore.

References

[1] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, and J. Lai, “Real-time RGB-D activity prediction by soft regression,” in *ECCV*, 2016, pp. 280–296.

[2] T. Lan, T.-C. Chen, and S. Savarese, “A hierarchical representation for future action prediction,” in *ECCV*, 2014, pp. 689–704.

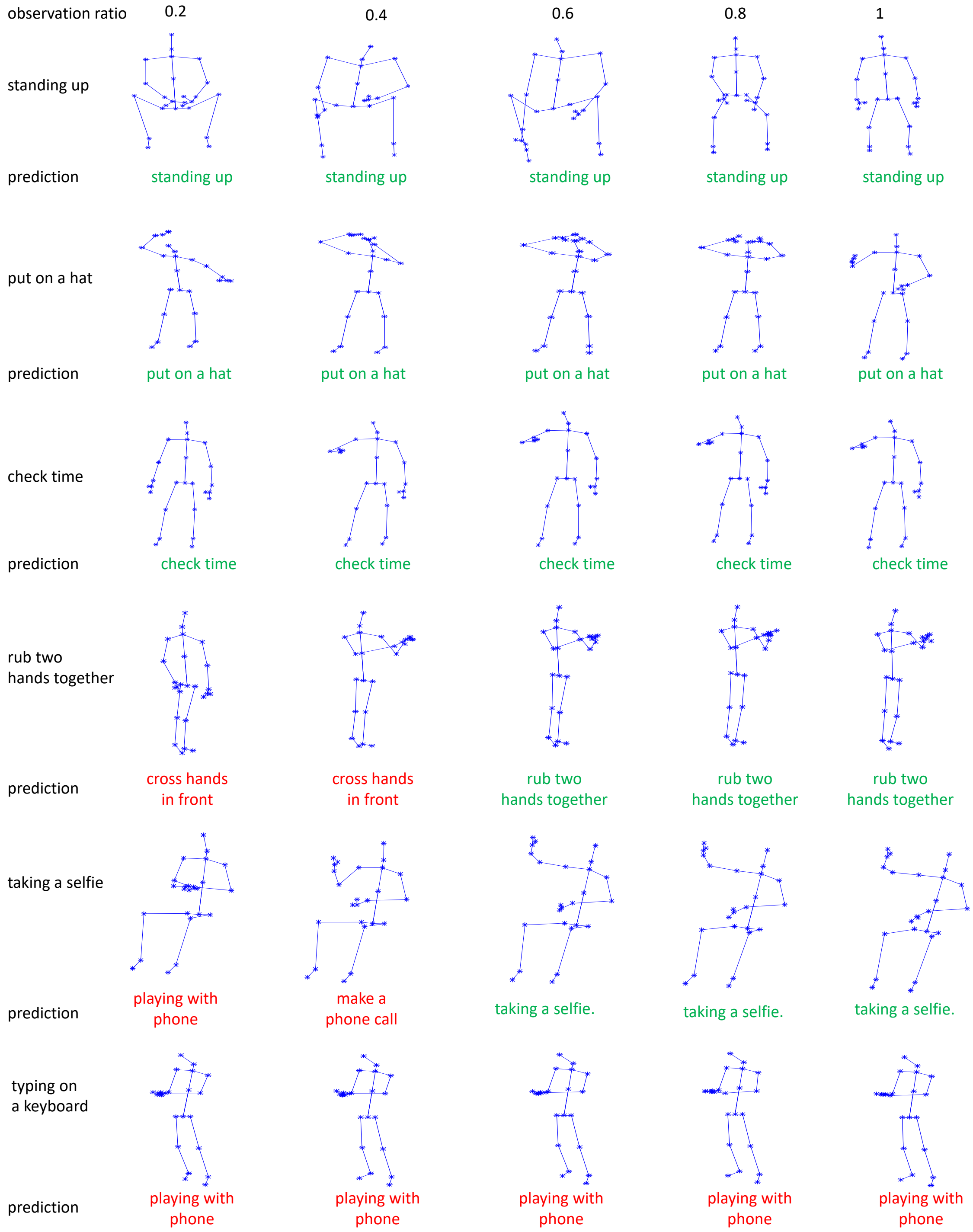


Fig. 6: Visualizations of action prediction at different observation ratios of each sequence. At observation ratio r, the input of the network is a partial sequence starting from the first frame to the rT^{th} frame (T denotes the number of frames of the full sequence). In this figure, only the last frame of the partial sequence rT is shown for simplicity.

[3] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, X. Xie *et al.*, "Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks." in *AAAI*, vol. 2, 2016, p. 8.

[4] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *CVPR*, 2012, pp. 1290–1297.

[5] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *CVPR*, 2014, pp. 588–595.

[6] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *CVPR*, 2015, pp. 1110–1118.

[7] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *CVPR*, 2016.

[8] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *ECCV*, 2016, pp.

816–833.

[9] Q. Ke, M. Bennamoun, S. An, F. Boussaid, and F. Sohel, "A new representation of skeleton sequences for 3D action recognition," in *CVPR*, 2017.

[10] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition."

[11] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: towards good practices for deep action recognition," in *ECCV*, 2016, pp. 20–36.

[12] Y. Wang and G. Mori, "Hidden part models for human action recognition: Probabilistic versus max margin," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1310–1323, 2011.

[13] D. Wu and L. Shao, "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 724–731.

[14] Y. Song, L.-P. Morency, and R. Davis, "Action recognition by hierarchical sequence summarization," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 3562–3569.

[15] A. Graves, "Neural networks," in *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012, pp. 15–35.

[16] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6645–6649.

[17] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *CVPR*, 2015, pp. 2625–2634.

[18] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-Based Human Action Recognition With Global Context-Aware Attention LSTM Networks," *TIP*, vol. 27, no. 4, pp. 1586–1599, 2018.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[20] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International Conference on Information Processing in Medical Imaging*. Springer, 2017, pp. 146–157.

[21] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1222–1230.

[22] S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, 2017, pp. 6670–6680.

[23] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[24] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4489–4497.

[25] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[26] J. Liu, A. Shahroudy, D. Xu, A. K. Chichung, and G. Wang, "Skeleton-based action recognition using spatio-temporal lstm network with trust gates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[27] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention lstm networks for 3d action recognition," in *CVPR*, 2017.

[28] Q. Ke, J. Liu, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Computer vision for human–machine interaction," in *Computer Vision for Assistive Healthcare*. Elsevier, 2018, pp. 127–145.

[29] J. Liu, N. Akhtar, and A. Mian, "Learning human pose models from synthesized data for robust rgb-d action recognition," *arXiv preprint arXiv:1707.00823*, 2017.

[30] H. Rahmani, A. Mian, and M. Shah, "Learning a deep model for human action recognition from novel viewpoints," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 667–681, 2018.

[31] H. Rahmani and M. Bennamoun, "Learning action recognition model from depth and skeleton videos," in *IEEE International Conference on Computer Vision*, 2017, pp. 5832–5841.

[32] M. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *ICCV*, 2011, pp. 1036–1043.

[33] Y. Kong, D. Kit, and Y. Fu, "A discriminative model with multiple temporal scales for action prediction," in *ECCV*, 2014, pp. 596–611.

[34] Q. Ke, M. Bennamoun, S. An, F. Boussaid, and F. Sohel, "Human interaction prediction using deep temporal features," in *ECCVW*, 2016, pp. 403–414.

[35] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Leveraging structural context models and ranking score fusion for human interaction prediction," *IEEE Transactions on Multimedia*, 2017.

[36] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena, "Recurrent neural networks for driver activity anticipation via sensory-fusion architecture," in *ICRA*, 2016, pp. 3118–3125.

[37] M. S. Aliakbarian, F. Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson, "Encouraging lstms to anticipate actions very early," in *ICCV*, 2017.

[38] Y. A. Farha, A. Richard, and J. Gall, "When will you do what?-anticipating temporal occurrences of activities," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2018.

[39] Q. Ke, M. Fritz, and B. Schiele, "Time-conditioned action anticipation in one shot," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9925–9934.

[40] Q. Ke, J. Liu, M. Bennamoun, H. Rahmani, S. An, F. Sohel, and F. Boussaid, "Global regularizer and temporal-aware cross-entropy for skeleton-based early action recognition," in *Asian Conference on Computer Vision*, 2018.

[41] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, and A. C. Kot, "Ssnet: Scale selection network for online 3d action prediction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8349–8358.

[42] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, and A. K. Chichung, "Skeleton-based online action prediction using scale selection network," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[44] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[45] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang, "Jointly learning heterogeneous features for RGB-D activity recognition," in *CVPR*, 2015, pp. 5344–5352.

[46] CMU, "CMU graphics lab motion capture database," in *http://mocap.cs.cmu.edu/*, 2013.

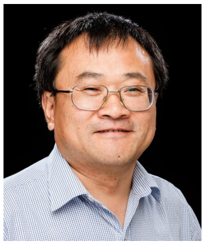
[47] Q. Ke, S. An, M. Bennamoun, F. Sohel, and F. Boussaid, "Skeletonnet: Mining deep part features for 3-d action recognition," *IEEE Signal Processing Letters*, vol. 24, no. 6, pp. 731–735, 2017.

[48] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Learning clip representations for skeleton-based 3d action recognition," *IEEE Transactions on Image Processing*, 2018.

Qiuhong Ke Qiuhong Ke received her Ph.D. degree from the University of Western Australia. She joined The University of Melbourne as a Lecturer (Assistant Professor) in January 2020. Before that she was a Postdoctoral Researcher at Max Planck Institute for Informatics. Her research interests include computer vision, machine learning, action recognition and action prediction.

Mohammed Bennamoun Mohammed Bennamoun is Winthrop Professor in the Department of Computer Science and Software Engineering at UWA and is a researcher in computer vision, machinedeep learning, robotics, and signalspeech processing. He has published 4 books (available on Amazon), 1 edited book, 1 Encyclopedia article (by invitation), 14 book chapters, 120+ journal papers, 250+ conference publications, 16 invited & keynote publications. His h-index is 48 and his number of citations is close to 11,000 (Google Scholar). He was awarded 65+ competitive research grants (approx. $17+ million in funding) from the Australian Research Council, and numerous other Government, UWA and industry Research Grants. He has delivered conference tutorials at major conferences, including: IEEE Computer Vision and Pattern Recognition (CVPR 2016), Interspeech 2014, IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP) and European Conference on Computer Vision (ECCV). He was also invited to give a Tutorial at an International Summer School on Deep Learning (DeepLearn 2017). He widely collaborated with researchers from within Australia (e.g. CSIRO), and internationally (e.g. Germany, France, Finland, USA). He served for two terms (3 years each term) on the Australian Research Council (ARC) College of Experts, and the ARC ERA 2018 (Excellence in Research for Australia).

Hossein Rahmani Hossein Rahmani received the BSc degree in computer software engineering from Isfahan University of Technology, Isfahan, Iran, in 2004, the MSc degree in software engineering from Shahid Beheshti University, Tehran, Iran in 2010, and the PhD degree from the University of Western Australia, in 2016. He has published several papers in top conferences and journals such as CVPR, ICCV, ECCV, and the IEEE Transactions on Pattern Analysis and Machine Intelligence. He is currently an associate professor (Lecturer) in the School of Computing and Communications at Lancaster University. Before that he was a research fellow in the School of Computer Science and Software Engineering, University of Western Australia. His research interests include computer vision, action recognition, 3D shape analysis, and machine learning.



Senjian An Senjian An received his B.S degree from Shandong University, the M.S. degree from the Chinese Academy of Sciences, and the Ph.D. degree from Peking University, China. He is currently a senior lecturer in the School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University. Before that he was a Research Fellow at the School of Computer Science and Software Engineering, The University of Western Australia. His research interests include machine learning, image processing, object detection and recognition.

Ferdous Sohel Ferdous Sohel received the PhD degree from Monash University, Australia, in 2009. He is currently an Associate Professor in Information Technology at Murdoch University, Australia. Prior to his joining Murdoch University, he was a Research Assistant Professor/Research Fellow at the University of Western Australia from 2008 to 2015. His research interests include computer vision, image processing, pattern recognition, multimodal biometrics, scene understanding, robotics, and video coding. He is a Member of Australian Computer Society and a Senior Member of IEEE.

Farid Boussaid Farid Boussaid received the M.S. and Ph.D. degrees in microelectronics from the National Institute of Applied Science (INSA), Toulouse, France, in 1996 and 1999 respectively. He joined Edith Cowan University, Perth, Australia, as a Postdoctoral Research Fellow, and a Member of the Visual Information Processing Research Group in 2000. He joined the University of Western Australia, Crawley, Australia, in 2005, where he is currently a Professor. His current research interests include neuromorphic engineering, smart sensors, and machine learning.