Non-local graph convolutional network for joint activity recognition and motion prediction

https://doi.org/10.1109/IROS51168.2021.9636107

Published in:
2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

Publisher rights
© 2021 IEEE
This work is made available online in accordance with the publisher’s policies. Please refer to any applicable terms of use of the publisher.

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person’s rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Open Access
This research has been made openly available by Queen's academics and its Open Research team. We would love to hear how access to this research benefits you. – Share your feedback with us: http://go.qub.ac.uk/oa-feedback
Non-local Graph Convolutional Network for joint Activity Recognition and Motion Prediction

Dianhao Zhang\textsuperscript{1}, Ngo Anh Vien\textsuperscript{2}, Mien Van\textsuperscript{1}, Sean McLoone\textsuperscript{1}

\textsuperscript{1}Queen's University Belfast, UK; \textsuperscript{2}Bosch Center for Artificial Intelligence, Germany

Abstract—3D skeleton-based motion prediction and activity recognition are two interwoven tasks in human behaviour analysis. In this work, we propose a motion context modeling that provides a new way to combine the advantages of both graph convolutional neural network and recurrent neural network for joint human motion prediction and activity recognition. Our approach is based on an LSTM encoder-decoder and a non-local feature extraction attention mechanism to model the spatial correlation of human skeleton data and temporal correlation among motion frames. The proposed network can easily include two output branches, Activity Recognition and Future Motion Prediction, and make them trained jointly to consolidate each other. Experimental results on Human 3.6M, CMU MoCap and NTU RGB-D dataset show that our proposed approach provides the best prediction capability among baseline LSTM-based methods, while obtaining a comparable performance with other state-of-the-art methods.

Keywords: LSTM, Graph Convolutional Network, Motion Prediction, Action Recognition, Human-robot Collaboration

I. INTRODUCTION

Human action recognition and motion prediction based on observations of their recent history of movement patterns and actions are vital capabilities that a robot must possess in order to achieve safe and seamless human-robot collaboration (HRC). For both 3D skeleton-based action recognition and motion prediction, the key is to have a representation of human skeleton data that enables the extraction of informative features. Action recognition enables a robot to decide on the corresponding actions, while the anticipation of a human’s future motion and position enables it to plan an execute its motions guaranteeing the safety of its human co-worker.

Developing a robotic system that can understand the circumstances under which it operates and reacts accordingly in a cooperative fashion is a challenging task. Especially for continuous HRC tasks containing multiple subtasks, it is necessary to enhance the context-aware skills of a robotic system by effectively modeling the historical human skeleton sequence. Recurrent and convolutional neural networks can extract features based on neighbour information in both the spatial and temporal domains. However, networks that focus on local interactions are often unable to adequately capture longer term dependencies. In this work, we address this first challenge by capturing non-local spatial-temporal contexts via an application of a Graph Convolutional Network (GCN) based attention mechanism to enhance the performance of Long Short-Term Memory (LSTM) based sequence-to-sequence (Seq2Seq) models [1] for human motion prediction and activity recognition.

The second challenge for HRC systems is its computational complexity which needs to be minimized to enable practical (low-cost) real-time implementation. As human action recognition and motion prediction are separate tasks, the computation time and memory requirements to execute these two tasks simultaneously can be high. One way to achieve a more efficient implementation is to jointly train this two tasks a single module so that they can consolidate each other’s prediction. We propose a non-local GCN-based (NGC) mechanism to provide a high-level shared feature map to be shared between these two task network branches. The feature extraction module is based on the previous observed human behaviour including both short-term and long-term information. Doing this, our GCN based non-local feature extractor can capture long-range dependencies directly by computing interactions between the hidden states of the encoder, regardless of the spatial distance between frames. The temporally long-range features will help improving the motion prediction module, hence as informative feedback to enhance the activity recognition module. In particular, our two decoder branches are: 1) the human motion trajectory prediction utilizes a LSTM layer with residual connections to predict future human poses, and 2) the action recognition module applies two feature extractors supported with feature augmentation and employs conditional random fields (CRFs) for activity classification.

Our main contributions can be summarized as:

- We propose a recurrent convolutional approach to capture long-term dependencies through which we can increase the predictable length of future time horizon based on non-local features. In this way we achieve an improved performance on both short-term and long-term human-robot interaction.
- We apply a GCN based relational LSTM to provide shared spatial-temporal feature extraction for both human action recognition and motion prediction.

II. RELATED WORK

A. Neural Networks on Graphs

Graph neural networks (GNN) [2] have received much attention recently as they can process data in irregular domains such as graphs or sets. For example, graph convolution networks (GCN) [3] is based on a fundamental convolution operation on the spectral domain. GCNs which have been proven to be effective for processing structured
data [4], hence widely been used to learning on structured skeleton data. Apart from graph recurrent neural networks (GRN) and graph convolutional neural networks (GCN), many alternative GNNs have been developed in the past few years, including graph autoencoders (GAEs) [5] and spatial-temporal graph neural networks (STGNNs) [4]. These learning frameworks can be built on GRN and GCN, or other neural architectures for graph modeling [4].

B. Skeleton-based Human Action Recognition

Many previous works have applied CNN or recurrent neural networks (RNN) [6], [7] for action recognition on skeleton-based data. However, these approaches are limited in their capacity to learn the complex, irregular and non-Euclidean structure of skeleton data. One solution to this problem is to split the whole human body and extract the components’ feature separately with subnetworks [8]. To further exploit the discriminative powers of different joints and frames, spatial-temporal attention [9] is applied to plug into networks to enable selectively focus on discriminative joints of the skeleton within one frame, and pay different levels of attention to the outputs at different time instances.

Recently, GNN or GCN have been demonstrated to be a more principled and effective choice for parsing the graph structure of skeleton data [10], which enables inner-correlation capture without segmentation of the whole body. Another example, TA-GCN [11], proposes to use an attention module in a GCN-based spatial-temporal model to extract more useful predictive features from graph data. However these works are focused on only action recognition so that it could not exploit available motion data to enhance the recognition ability. In addition, to prevent the loss of information on the correlation between human body joints during training, a strategy called neural architecture search (NAS) was employed at each iteration to enhance GCN-based human action recognition [12].

C. Skeleton-based Human Motion Prediction

There are a number of examples showing accurate motion prediction would improve task performance. For instance, robust and fast human motion prediction in HRC can minimize robot response time and latency.

With the advances in deep learning, RNN based approaches such as LSTM have been shown to be a powerful tool for motion prediction in recent years [13]. A recent common strategy for motion prediction is to use a recurrent neural network (RNN) to encode temporal information [1], [14]. Although sequence-to-sequence models perform well for short-term prediction, encoding of long-term historical information is challenging [15]. To account inter-dependency between connected human body components, graph based approaches are applied to structurally learn dynamics of different components’ property. Spatial-temporal graph neural networks (STGNNs) have been proposed to model such systems [16] considering inner spatial-temporal information.

III. SPATIAL-TEMPORAL GRAPH NEURAL NETWORK BASED HUMAN FEATURE LEARNING

As a summary, our network architecture will consist of following modules:

1) GCN-LSTM Encoder which models human motion dynamics by encoding the temporal evolution of skeletal graph data in an encoding vector. We employ GCN layers to extract features from raw skeletal graphs, then followed by bi-directional LSTM (Bi-LSTM) layers to learn past and future contexts within a sequence of $\tau$ poses.

2) Non-local GCN Module which bridges the encoded state of the GCN-LSTM Encoder with the action recognition and motion prediction modules. We employ residual blocks (Res-GCN) [17] to extract spatial features from each frame, while the GCN-based non-local network models the relations between the current hidden state at time $t$ and the hidden states from $t = 1$ to $t = \tau$. It works like a self-attention module.

3) Two Task-specific Decoders: The first LSTM-based decoder is designed to predict the future sequence of skeletal graph poses (human motion prediction). The second CNN-CRF-based decoder performs action recognition.

A. GCN-LSTM Encoder

The overall structure and information flow of our encoder module are shown in Fig. 1. In the first layer of the encoder, the skeleton graph is embedded in a vector $E \in \mathbb{R}^{N \times 3}$ by a fully connected (FC) layer, where $N$ denotes the number of human joints. The input to the embedding layer is $X_{prev}$, where $X_{prev} = \{X^{(1)}, \cdots, X^{(\tau)}\} \in \mathbb{R}^{T \times N \times 3}$ is a sequence of $\tau$ skeletal graphs. This is then fed into a GCN layer to extract spatial graph feature $G$. Then we apply 4-layer Bi-LSTM to generate output $O$ and hidden states $H$. Denoting $f_{em}, f_{GCN}, f_{Bi-LSTM}$ as the embedding, GCN and Bi-LSTM encoder function, respectively. The operation of the encoder module can be formulated mathematically as:

\[
E = f_{em}(X_t; \theta_{emb})
\]

\[
G = f_{GCN}(E; \theta_{GCN})
\]

\[
O, H = f_{Bi-LSTM}(G; \theta_{Bi-LSTM})
\]

where $E = \{e_1, \ldots, e_\tau\}$ is the matrix of embedded pose feature output from the fully connected layer for the observed sequence of $\tau$ skeleton frames. Parameters $\theta_{emb}, \theta_{GCN}$ and $\theta_{Bi-LSTM}$ are the trainable parameters of the fully connected embedding layer, GCN layer and Bi-LSTM layer, respectively.

B. Non-local GCN Module

In this work, we employ two GCN layers that are based on residual blocks (Res-GCN) [17] and the self-attention mechanism [18] to further extract non-local features from output of the encoder, as described in Fig. 2. The attention block processes three inputs, a query $Q$, keys $K$, and values $V$ generated from the hidden states and output of the encoder block. The output of the self-attention function is defined as:

\[
Att(Q, K, V) = \frac{1}{Z} \text{softmax}(QK^T)V,
\]
where $Z$ is a normalization factor. The query and the key are the hidden states within observed time length $\tau$, $Q, K \in \mathbb{R}^{T \times N \times D_h}$ and $N_h = N \times D_h$ is the hidden size of the Bi-LSTM layer from the encoder. The value $V$ is the output from encoder, $V \in \mathbb{R}^{T \times N \times 2D_h}$, where $N_{out} = N \times 2D_h$ denotes the output size of the encoder. The spatial-temporal correlation between $Q$ and $K$ is calculated and added into the generated feature map after multiplying with $V$.

In GCN, the input is modeled as a fully-connected graph with $N$ nodes. Firstly, a Res-GCN takes as input a matrix $Q \in \mathbb{R}^{T \times N \times D_h}$. Given this information and a set of trainable weights $W(t) \in \mathbb{R}^{N_h \times N_g}$, where $N_g = N \times D_g$ is the output dimension of the Res-GCN layer. A temporal convolution operation is employed to do temporal feature extraction after Res-GCN layer, the process of ResGCN-TC to get $Q$ can be described as:

$$g(H) = \text{ReLU}(f_{bn}(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}HW)),$$

$$f_{RG}() = \text{ReLU}(f_{bn}(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}g()W)),$$

$$f_{TC}() = \text{ReLU}((\text{Conv}(\text{ReLU}(\text{Conv}())))),$$

where $f_{bn}()$ is the batch normalization function and $Q = f_{TC}(f_{RG}(H))$ is the Res-GCN function to achieve query $Q$. The functions $f_{RG}$ and $f_{TC}$ denote the Res-GCN function and temporal convolution function, respectively. Res-GCN module consists of two GCN layers, each followed with batch normalization and ReLU activation. While TC module consists of two CNN layers, each followed by a ReLU activation function, as described in Fig. 2. The term $D$ is the diagonal node degree matrix used to normalize the adjacency matrix $A$, $\text{ReLU}(\cdot)$ is the ReLU activation function. In practice, we use a symmetric normalization $D^{-\frac{1}{2}}A D^{-\frac{1}{2}}$ to avoid merely averaging of neighboring nodes. In this work, $\hat{A} = A + I$ to add self-loop, where $I$ is the identity matrix and $\hat{D}$ is the diagonal node degree matrix of $\hat{A}$. After the Res-GCN operation, $Q$ and $K$ will be transformed to dimension $T \times N \times D_g$. A TC layer takes convolutional operation over the temporal dimension to generate spatial-temporal feature map. The generated spatial-temporal feature matrix multiplies with each other and then go through softmax function to capture spatial-temporal correlation between hidden states. We can describe this non-local feature extraction process as:

$$f = f_{TC}(f_{RG}())$$

$$P = \text{softmax}(f(H) \otimes f(H)^T) \otimes f_{RG}(O)$$

C. Activity Recognition in Human Robot Collaboration

Given the sophisticated attention feature map $P$ in Eq. 6 extracted from observed human skeleton data, we now propose the first decoder to classify activity. We first augment this feature map by incorporating the temporal semantics extracted by a Res-CNN layer. In detail, the extracted semantic feature from spatial-temporal frame index sequence with Res-CNN is added on further extracted feature map (processed by two NGC module). Then we apply the spatial temporal pooling (SMP) operation on the generated feature to obtain a temporal feature representation. Then two temporal convolutional neural network layers combined with max-pooling are employed to learn feature representation for classification. In this work, we apply CRF for label sequence estimation [19]. We try to model the conditional probability of a label sequence $Y$, given an extracted attention feature of observed frames, $P$, i.e. CRFs model $P(Y|P)$. The CRF can describe the conditional probability as:

$$P(Y|P) = \frac{1}{Z(P)} \prod_{t=1}^{T} \Psi_f(y_{t-1}, y_t, P_t)$$

where $\tau$ denotes the observed time length, $\Psi_t$ is the weight function on the transition from state $y_{t-1}$ to state $y_t$ when the current observation is $P$. Feature map $P$ contains the
full body features in observed frames \( \{1, \ldots, \tau\} \), while \( Z \) is a normalization function. The goal of the proposed model, as depicted in Fig. 3, is for a given feature map \( P \) and the model’s parameter vector \( \theta \), to find the most probable label \( y_t \) by maximizing the conditional probability \( P(y_t | P; \theta) \). This procedure is implemented by a Viterbi decoder to estimate the probability of the most probable path ending.

D. NGC Attention based LSTM Encoder-Decoder for Human Motion Prediction

The output of NGC feature extractor represents a context that can be used as the initial hidden state to the motion prediction decoder. The first input sequence to the decoder is the first frame in \( X^{(\tau+1)} \), \( f(0) \). In the subsequent steps of decoding, \( f^{(t-1)} \) could be the decoder’s own prediction of the previous step or the ground-truth of the previous step. The teacher forcing method is applied for addressing slow convergence and instability when training recurrent networks. Based on that input representation, the decoder generates an output sequence that represents a sequence of future motion.

Our LSTM based motion prediction module, as depicted in Fig. 4, is inspired by the Seq2Seq network [20], so we name it NGC-Seq2Seq. It computes future motion predictions as \( X_{\text{fut}} = f_{\text{pred}}(X_{\text{prev}}; \theta_{\text{pred}}) \), where we denote \( f_{\text{pred}}(.) \) the motion prediction module, \( X_{\text{fut}} = [X^{\tau+1}, \ldots, X^T] \) is the future motion sequence; and \( \theta_{\text{pred}} \) denotes the trainable parameters of this prediction module. In particular, to produce the \((t+1)^{th}\) pose \((t \geq 0)\) the motion prediction module works as follows,

\[
H^{(\tau+1)}_F = f_{\text{LSTM}}(\Delta P; \theta_{\text{pred}}),
\]

\[
\hat{X}^{(\tau+1)}_{\text{fut}} = f_{\text{FC}}(\hat{H}^{(\tau+1)}_F) + P,
\]

\[
\hat{X}^{(\tau+2)}_{\text{fut}} = \hat{X}^{(\tau+1)}_{\text{fut}} + f_{\text{LSTM}}(\Delta \hat{X}^{(\tau+1)}_{\text{fut}}; \theta_{\text{pred}})
\]

where \( f_{\text{LSTM}}(.) \) and \( f_{\text{FC}}(.) \) represent LSTM decoder predictor and fully connected layers (FC), respectively. As we are predicting an offset between frames, the term \( \Delta X^{(t)} \) denotes the displacement of the current human motion sequence at time \( t \), \( \hat{H}^{(t)}_{\text{fut}} \) is the hidden state of LSTM at time \( t \), and \( \Delta P \) denotes the displacement of the attention feature map between the current \( \tau \) and the previous \( (\tau - 1) \) prediction step. Firstly, we feed the updated hidden states and current feature displacement into the LSTM decoder to produce the features that reflect future displacement. The first input to the LSTM decoder is the last output from the attention feature \( P \). Then, we adopt the predicted displacement over the previous pose to predict the next frame. One limitation for the LSTM decoder is, during inference, the previous true target tokens are replaced by tokens generated from the model’s hidden state [21]. To solve this problem for better long-term anticipation performance, we apply teacher forcing as a learning strategy to diminish the influence from the model’s own error [22]. At every time step, the choice on ground truth or its own prediction is determined by a coin flipping probability \( p \). Initially, \( p = 1 \) (i.e. teacher forcing), and it decays exponentially with a factor \( \beta = 0.995 \) per epoch.

IV. EXPERIMENTS AND ANALYSIS

A. Implementation Details and Metrics

The models are implemented with PyTorch. The size of input to encoder is \( N \times 3 \), and the hidden size of Four-layer Bi-LSTM is \( N \times 8 \). In NGC module, for the channels receiving hidden states, the first GCN layer in Res-GCN convert spatial dimension from \( N \times 8 \) to \( N \times 4 \), while the second GCN layer convert spatial dimension from \( N \times 4 \) to \( N \times 3 \). As the output from encoder is the concatenation of forward and backward hidden states, in terms of the Res-GCN on the bottom channel, the first GCN layer convert spatial dimension from \( N \times 16 \) to \( N \times 8 \), while the second GCN layer convert from \( N \times 8 \) to \( N \times 3 \). The TC in the bottom channel is the same as the ones in previous channels. The kernel size of temporal convolution is \( 1 \times 1 \). The first layer of CNN in TC convert temporal dimension from \( \tau \) to 64, while the second CNN layer convert temporal dimension from 64 to \( \tau \).

In motion prediction decoder, the hidden size of the LSTM decoder is \( N \times 8 \) and the fully connected layer convert the spatial dimension back to \( N \times 3 \), which can be formed as skeleton data. In action recognition decoder, the Res-CNN used to process temporal semantics convert temporal dimension from \( \tau \) to 64, then back to \( \tau \). The two-layer
CNN between spatial-temporal max-pooling convert spatial dimension from $N \times 8$ to $N \times 16$. The loss used for joint angles is the $l_1$ average distance between the ground-truth joint angles, and the predicted ones. We apply Huber loss [23] as the prediction loss. We also include a penalty term in the Huber loss, $p = \omega (x^2 + y^2 + z^2 - 1)^2$, where $x$, $y$ and $z$ are the 3D coordinates of the predicted and ground truth skeleton. Assuming the $i^{th}$ predictions and ground truth be $(X_{fut})_i$ and $(\hat{X}_{fut})_i$, for $N$ samples in one mini-batch, the prediction loss $L_{pred}$ can be expressed as:

$$L_{pred} = \frac{1}{N} \left\{ \begin{array}{ll}
0.5 \frac{(X_{fut}^i - \hat{X}_{fut}^i)^2}{\beta} & \text{if } X_{fut}^i - \hat{X}_{fut}^i < \beta \\
|X_{fut}^i - \hat{X}_{fut}^i| - 0.5 \beta & \text{otherwise}
\end{array} \right. $$

For action recognition, the conditional random field are commonly trained by maximizing the conditional log likelihood of a labeled training set to estimate the weight vector. Let the true label of the $n^{th}$ training sample be $Y_n$ and the estimated be $\hat{Y}_n$. As the generation of loss function contains the computation of the arg max function. For $N$ training samples in one mini-batch, the action recognition loss is formulated as:

$$L_{rec} = \min_{\theta} \left\{ \frac{\lambda}{2} \sum_{n=1}^{N} L_{i}(Y, \hat{Y}) \right\}, \quad (11)$$

where the factor $\lambda$ is a trade-off constant, which can be exploited to provide a regularization balance term between the model complexity and fitting date used to avoid or reduce over-fitting in model learning. Parameter $\theta$ is the trainable parameter of the CRF model. Inspired by [24], the function $L_n(Y, \hat{Y})$ denotes the measure cost of the wrong estimation for the $n^{th}$ training sample, which is expressed as:

$$L_n(Y, \hat{Y}) = \sum_{t=1}^{T-\tau} (1 - \delta(Y^n_t, \hat{Y}^n_t))$$

where $\delta(Y^n_t, \hat{Y}^n_t) = \begin{cases} 
1 & Y^n_t = \hat{Y}^n_t \\
0 & \text{otherwise}
\end{cases}$

We can jointly train the motion prediction and activity recognition modules with a combined loss function as:

$$L = \lambda L_{pred} + (1 - \lambda) L_{rec}$$

where $\lambda$ is a trade-off value to balance the importance of two tasks. We have carried out grid-search to find the best $\lambda$, and found the best $\lambda = 0.4$ having the best validation loss. We use Adam optimizer to train our model, where the learning rate is initially 0.001 and decays by 10 every 10 epochs. The model is trained with batch size 16 for 100 epochs on NVIDIA Quadro P4000 GPU.

B. Datasets

a) Human 3.6M dataset: Human3.6M is a dataset containing 3.6 Million accurate 3D Human poses of motion [25]. The dataset consists of 3.6 million different human poses collected. Worth to be mentioned, the skeleton data in this dataset containing 32 joints.

<table>
<thead>
<tr>
<th>Methods</th>
<th>CS</th>
<th>CV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lie Group [27]</td>
<td>50.1%</td>
<td>52.8%</td>
</tr>
<tr>
<td>H-RNN [28]</td>
<td>59.1%</td>
<td>64.0%</td>
</tr>
<tr>
<td>Deep LSTM [26]</td>
<td>60.7%</td>
<td>67.3%</td>
</tr>
<tr>
<td>PA-LSTM [26]</td>
<td>62.9%</td>
<td>70.3%</td>
</tr>
<tr>
<td>ST-LSTM+TS [29]</td>
<td>69.2%</td>
<td>77.7%</td>
</tr>
<tr>
<td>Temporal Conv [30]</td>
<td>74.3%</td>
<td>83.1%</td>
</tr>
<tr>
<td>Visualize CNN [31]</td>
<td>76.0%</td>
<td>82.6%</td>
</tr>
<tr>
<td>ST-GCN [4]</td>
<td>81.5%</td>
<td>88.3%</td>
</tr>
<tr>
<td>DPRL [32]</td>
<td>83.5%</td>
<td>89.8%</td>
</tr>
<tr>
<td>SR-TSL [33]</td>
<td>84.8%</td>
<td>92.4%</td>
</tr>
<tr>
<td>HCN [34]</td>
<td>86.5%</td>
<td>91.1%</td>
</tr>
<tr>
<td>STGR-GCN [35]</td>
<td>86.9%</td>
<td>92.3%</td>
</tr>
<tr>
<td>motif-GCN [36]</td>
<td>84.2%</td>
<td>90.2%</td>
</tr>
<tr>
<td>AS-GCN [10]</td>
<td>86.8%</td>
<td>94.2%</td>
</tr>
<tr>
<td>AGC-LSTM [37]</td>
<td>88.5%</td>
<td>95.1%</td>
</tr>
<tr>
<td>DGNN [38]</td>
<td>89.2%</td>
<td>95.0%</td>
</tr>
<tr>
<td>Sym-GNN [39]</td>
<td>90.1%</td>
<td>96.4%</td>
</tr>
<tr>
<td>NGC-CRF</td>
<td>95.2%</td>
<td>98.5%</td>
</tr>
</tbody>
</table>

TABLE I: Comparison of action recognition on NTU-RGB+D. The accuracy on both Cross-Subject (CS) and Cross-View (CV) benchmarks.

b) NTU RGB+D Dataset: NTU-RGB+D was proposed by Amir Shahroudy et al. [26] in 2016. The proposed dataset consists of 56, 880 RGB+D video samples, captured from 40 different human subjects. The dataset included RGB videos, depth sequences, skeleton data (3D locations of 32 major body joints), and infrared frames. This dataset contains both human motion and labeled activity. NTU RGB+D also provides two standard test methods: Cross subject (CS) and Cross view (CV).

c) CMU Mocap: CMU Mocap dataset is collected in the lab contains 12 Vicon infrared MX-40 cameras. The skeleton data in CMU contains 41 joints. There are 144 human motion categories in this dataset. We test our motion prediction model on the dataset in the same way as that in Human 3.6M.

C. 3D Skeleton-based Action Recognition

For action recognition, we first show the classification accuracy of our method in comparison with baselines on two benchmarks of NTU-RGB+D, i.e. Cross-Subject (CS) and Cross-View (CV). Table I presents recognition accuracy of methods, it shows our proposed method outperforms all baselines on both benchmarks. Note that our network is trained to exploit the motion prediction module, that means it uses both human motion and labeled activity data.
D. 3D Skeleton-based Motion Prediction

To validate the proposed model, we show the prediction performance for both short-term and long-term motion prediction on Human 3.6M (H3.6M). We quantitatively evaluate various methods by the mean angle error (MAE) between the generated motions and ground-truths in angle space. Note that we just need to ignore the action recognition decoder.

a) Short-term motion prediction: Short-term motion prediction aims to predict the future poses within 400 milliseconds. We train our model to generate future 10 frames for each input sequence on Human 3.6M. The results in Table II show that our model can capture long-term dependencies to achieve more stable short term prediction. However, the performance to anticipate the most recent frames is a bit worse than the ones of Imit-L, Traj-GCN, and Sym-GNN. This might explain that one can not have the best compromise between long and short-term dependencies in a single model. It is very encouraging that our model provides the best performance within the same future time length given the same length of a previous human motion sequence.

b) Long-term motion prediction: Long-term motion prediction aims to predict the poses within 1000 milliseconds. In this work, we compare the long-term performance with state-of-the-art baselines on CMU Mocap dataset in Table III. The experimental results show that the proposed model substantially exceeds the baselines in both short-term and long-term prediction.

V. ABLATION STUDY

A. Mutual Effects of Prediction and Recognition

As motion prediction and action recognition are trained simultaneously, a combined loss Eq.18 is presented to train the HRC system. In order to achieve action category and predicted human pose simultaneously, we train the model on NTU-RGB+D dataset under different $\lambda$, where $\lambda$ is a trade-off value applied to show the influence proportion of motion prediction and action recognition. By randomly shuffling a percentage, we can decide the best balance of these two modules. Table IV presents the recognition accuracy and average MAEs for short-term prediction on NTU RGB+D and Human 3.6M respectively. The results suggest that proper combination of prediction and recognition is able to support each other to generate more stable HRC system. According to Table IV, $\lambda = 0.4$ is most suitable for both recognition and prediction. Compared to model trained with separate loss function, combined loss can achieve better performance and enable simultaneously training.

B. Predicted Sequence

In order to prove NGC-Seq2Seq provides better prediction than baselines, we compare the generated future poses of "discussion" from different models (i.e. Res-sup, Traj-GCN and NGC-Seq2Seq). According to Fig 5, We found that Res-sup cannot predict dynamics of body components. Prediction errors also exist in Traj-GCN model after 800 ms. Although NGC-Seq2Seq model has slightly difference compared with ground truth after 800 ms, the difference value is acceptable and it is close to the safety target. Worth to mention, we only apply joint-scale graph in training.

C. Statistical Significance

In order to prove the significance of Bi-LSTM and NGC attention on action recognition and long-term prediction, a t-test was applied to the model without Bi-LSTM and NGC based on NTU-RGBD dataset. The t-test is a commonly used statistical test which evaluates whether the difference between two sets of data is random or statistically significant. Table V illustrates the average accuracy and MAE over 10 runs between model without Bi-LSTM and NGC attention. Also, we divide the NTU-RGBD dataset into two random folds and run 10 times. Then, the results are collected, and the t statistic is calculated. Here, p-value on action recognition and long-term motion prediction both close to zero, which suggests that the differences in performances between the two sets of dataset were found to be statistically significant. P-values very close to zero indicate that the confidence of the evaluation is higher than 99%, because $p \ll 0.05$. Hence, the proposed approach offers a significant increase in the performance of action recognition and motion prediction. What’s more, NGC attention and Bi-LSTM are proved to be important to action recognition and long-term motion prediction.

VI. CONCLUSION

In this paper, the proposed non-local graph convolutional (NGC) network model helps to solve the challenges of long-range dependencies capture and shared feature map for both human action recognition and motion anticipation. The human motion prediction module consists of NGC module and a LSTM decoder. The activity recognition module contains
a couple layers of NGC and then scoring by conditional random fields. Both results for prediction and recognition proves that the proposed model can capture inner spatial-temporal correlation in a task and long term dependencies, which achieves a consistent improvement over the existing methods. In the future, we will incorporating the predicted human state as reference to the robot controller to enhance the safety and minimize the total time of HRC.

**REFERENCES**


[34] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence Feature Learning from Skeleton Data for Action Recognition and Detection with Hierarchical Aggregation, 2018.


