MSR-GCN: Multi-Scale Residual Graph Convolution Networks for Human Motion Prediction

Lingwei Dang[1], Yongwei Nie[1*], Chengjiang Long[2], Qing Zhang[3] and Guiqing Li[1]

[1]School of Computer Science and Engineering, South China University of Technology, China
[2]JD Finance America Corporation, USA
[3]School of Computer Science and Engineering, Sun Yat-sen University, China

Abstract

Human motion prediction is a challenging task due to the stochasticity and aperiodicity of future poses. Recently, graph convolutional network (GCN) has been proven to be very effective to learn dynamic relations among pose joints, which is helpful for pose prediction. On the other hand, one can abstract a human pose recursively to obtain a set of poses at multiple scales. With the increase of the abstraction level, the motion of the pose becomes more stable, which benefits pose prediction too. In this paper, we propose a novel multi-scale residual Graph Convolution Network (MSR-GCN) for human pose prediction task in the manner of end-to-end. The GCNs are used to extract features from fine to coarse scale and then from coarse to fine scale. The extracted features at each scale are then combined and decoded to obtain the residuals between the input and target poses. Intermediate supervisions are imposed on all the predicted poses, which enforces the network to learn more representative features. Our proposed approach is evaluated on two standard benchmark datasets, i.e.*, the Human3.6M dataset and the CMU Mocap dataset. Experimental results demonstrate that our method outperforms the state-of-the-art approaches. Code and pre-trained models are available at: https://github.com/Droliven/MSRGCN.*

1. Introduction

Human motion prediction plays a critical role in many fields, such as human-computer interaction, autonomous driving, and video completion. Simple periodic motion patterns can be tackled by traditional methods such as hidden Markov model [3], linear dynamic system [37], restricted Boltzmann machine [44], Gaussian process latent variable models [46] and random forests [22], while more complex motion is intractable for these methods. The latest approaches are almost all data-driven methods with deep learning. However, considering the stochasticity and aperiodicity of human motion, it remains a challenging task to produce accurate motion in long term giving observed arbitrary poses without an action label. The main difficulty is to model the spatiotemporal dependencies of human poses.

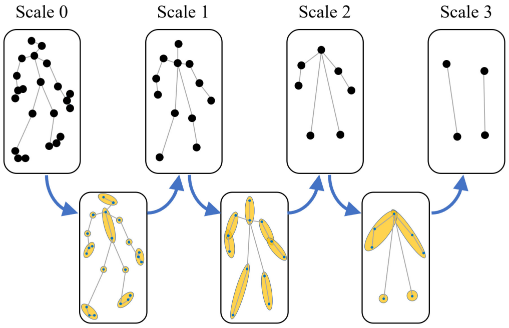

Figure 1. A human pose can be abstracted step by step to obtain a series of poses from fine to coarse scale, by grouping joints in close proximity together and replacing the group with a single joint.

Lots of prior efforts with Convolutional Neural Networks (CNNs) [50, 27], Recurrent Neural Networks (RNNs) [8, 35, 42, 43, 38, 10, 5, 2], and Generative Adversarial Networks (GANs) [54, 9, 19, 11, 6, 45, 21], have been made for tackling the challenging task. However, they neglect the inner-frame kinematic dependencies between body joints. Although they have achieved success in some cases, the prediction accuracy still heavily depends on the size of convolution filters and the stability of the frame-by-frame pre-

[*]Corresponding author: nieyongwei@scut.edu.cn

This research is sponsored in part by the National Natural Science Foundation of China (62072191, 61802453, 61972160), in part by the Natural Science Foundation of Guangdong Province (2019A1515010860, 2021A1515012301), and in part by the Fundamental Research Funds for the Central Universities (D2190670).

diction. Nowadays, Graph Convolution Networks (GCNs) have been widely used in various fields as well as in the task of human motion prediction [34, 26, 7, 24, 28, 53, 40], which work very well for non-grid graph-structured data especially skeleton-based 3D human pose sequences. Recently, Mao *et al*. [34] jointly model spatial structure by GCNs with learnable connectivity and temporal information via discrete cosine transformation (DCT) to predict human motion. Li *et al*. [26] propose a dynamic multi-scale graph neural network with an encoder-decoder framework to extract deep features at multiple scales. Although these two works exhibit promising results on benchmark datasets, it is still far from satisfactory to handle complex scenarios.

In this paper, we propose a Multi-Scale Residual Graph Convolution Network named MSR-GCN, as illustrated in Figure 3, for 3D human motion prediction. By treating a human pose as a fully connected graph whose vertices are the pose joints, we can employ a graph convolution network to dynamically learn the relations between all pairs of joints flexibly regardless of the physical distance between them. But GCN alone cannot capture the hierarchical structure of human pose [34]. That is, as shown in Figure 1, one can abstract a human pose by grouping joints in close proximity together and representing the group by just one joint, to obtain a coarser pose. Since a group of joints usually come from the same body part, gradually abstracting body parts in this way can significantly stabilize the motion pattern of the body. Li *et al*. [26] have also observed this natural hierarchical structure of human pose, but they aim to extract rich features using multi-scale joint abstraction and decodes the future poses with a recurrent decoder fed with the extracted multi-scale features. In contrast, the encoder and decoder in our method are organized in a U-Net-like multi-scale manner equipped with intermediate losses.

Based on the above analysis, we compensate GCN with the capacity of modeling hierarchical and contextual information of human pose by designing multiple GCNs with a multi-scale architecture. A group of the GCNs forms a descending path to extract features from fine to coarse scale, followed by another group of GCNs that extract multi-scale features inversely along an ascending path. Based on these features, we predict poses at all scales and impose intermediate supervision for more representative features. We also add residual connections between the input and the output poses, as shown in Figure 2, making the whole framework learn residuals instead of the target poses directly.

In short, our main technical contributions are as follows:

- We propose a novel multi-scale residual Graph Convolution Network (MSR-GCN) for human pose prediction in an end-to-end manner, which consists of multiple GCNs organized in a multi-scale architecture.
- The well-designed descending and ascending GCN blocks are able to ensure the feature extraction in both fine-to-coarse and coarse-to-fine manners.
- The intermediate supervision enforces the feature extraction stage to learn more representative features at multi-scale, which are then fed together to the corresponding end/decoding GCN with residual connections to ensure high quality for the final prediction.

Experiments are conducted on two benchmark datasets, *i.e.*, the Human3.6M dataset and the CMU Mocap dataset, which show our MSR-GCN outperforms the state-of-the-art methods in both short-term and long-term.

2. Related work

Human motion prediction. Many deep learning based methods have been proposed to handle human motion prediction. Existing CNN-based works like [50, 27] treat the pose sequence as a two-dimensional matrix where one axis is the spatial axis and another one indicates the temporal axis, then spatiotemporal convolutional filters can be used to the pose data like what has been done for an image. However, pose data, in essence, is very different from images, lacking repeated elements that give a high response to the same filter, thus reducing the effectiveness of the convolutions. Although RNN-based methods like [8, 35, 42, 43, 38, 10, 5, 2] have advantages in dealing with time-related tasks, the discontinuity and error accumulation problems often happen because of the frame-by-frame prediction manner. Also, the training of RNN models is easy to collapse with gradient explosion or disappearing. What's more, these networks neglect the inner-frame kinematic dependencies between body joints. Generative adversarial networks [54, 9, 19, 11, 6, 45, 21] are deemed to generate realistic data whose pattern is similar to the training data. Nevertheless, they are vulnerable and require skillful training. Transformer-based networks like [4, 1] are supposed to be capable of capturing long-range temporal dependencies directly but usually have quite high computing cost. Unlike above-mentioned methods, our proposed MSR-GCN can model the inner-frame dependencies between body joints in the graph structure.

Graph Convolution Networks (GCNs) are suitable for tasks with non-grid and graph-structural data, *e.g*., biological gene, point cloud, human social network [49], and human motion prediction for the graph-structure nature of the human skeleton. They have been successfully applied to many applications like visual recognition [12, 14, 13, 15, 30, 32, 31, 16, 29], object detection [47, 33, 17], action localization [51, 18], and trajectory prediction [39]. In particular, since graph convolution is more inclined to capture spatial information, Si *et al*. [41] combines it with LSTM to enhance its capability of modeling temporal dependencies between human skeleton connections. Works of [34, 25, 7]

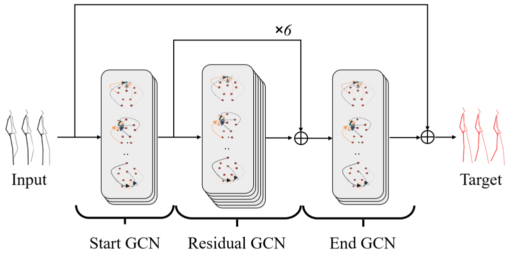

Figure 2. The basic GCN model for pose prediction comprising a start GCN, 6 residual GCNs, and an end GCN. The start GCN maps the input from pose space to feature space, the residual GCNs are used to extract features in the feature space, and finally, the end GCN maps the features back to the poses. A residual connection is added between the input and output poses, making the whole network learn residuals rather than the target poses directly.

allow graph convolution network to learn relations between any pair of human joints. Mao *et al.* [34] design a fully connected GCN to adaptively learn the necessary connectivity for the motion prediction task and apply discrete cosine transformation (DCT) to handle temporal information. Cui *et al.* [7] enhance the role of natural connectivity of human joints among all the edges of the fully connected graph. Li *et al.* [26] propose a graph neural network with a multi-scale graph computational unit where features are extracted at a single individual scale and then fused across scales. Differently, we use GCNs at different scales to extract features for these scales separately.

3. Methodology

Human pose prediction is a task to produce future pose sequence given the currently observed frames. Supposing the historical poses are $X_{1:T_h} = [X_1, ..., X_{T_h}] \in \mathbb{R}^{J \times D \times T_h}$ with T_h frames, among which X_{T_k} depicts a single 3D human pose with J joints in the D-dimensional space (here D is 3) at time t. Similarly, the future pose sequence with T_f frames is defined as $X_{T_h+1:T_h+T_f}$. We need a model $\mathcal{F}_{predict}(\cdot)$ to predict the future unknown pose sequence $\hat{X}_{T_h+1:T_h+T_f}$ giving $X_{1:T_h}$ that approximates the ground truth $X_{T_h+1:T_h+T_f}$ as much as possible. We fulfill this task by proposing a novel Multi-Scale Residual Graph Convolution Network called MSR-GCN, as illustrated in Figure 3.

In the following, the basic GCN model for pose prediction is introduced firstly, then the multi-scale architecture used to obtain superior prediction accuracyis is shown.

3.1. Basic GCNs

Firstly, we reformulate our prediction objective by rearranging the input and output pose sequences. Instead of performing prediction based on $X_{1:T_h}$, we replicate the last pose X_{T_h} for T_f times, obtaining a sequence of length $T = T_h + T_f$. We then use this sequence as the input to predict the future pose sequence comprising of $\hat{X}_{1:T_h}$ and $\hat{X}_{T_h+1:T_h+T_f}$. According to [34], this prediction task can be translated to compute a residual vector, which we also find very effective to improve the prediction accuracy.

For pose prediction, it has been proven very useful to model the spatial structure of the poses [34, 7]. This is because the spatial dependencies between human joints exhibit inherent and consistent characteristics over the whole action period, which is of great importance for human pose prediction. The dependencies that can be utilized are not confined to joints with kinematic links such as between elbow and wrist, but any pair of joints can affect each other. For example, when a person walks, the hands vibrate periodically, so it is essential to explore the dependencies of two hands for their predictions. GCN [20] is good at discovering these relationships by viewing a pose as a fully-connected graph with K nodes, where $K = J \times D$, and an adjacency matrix $\mathbf{A} \in \mathbb{R}^{K \times K}$ which represents the strength of edges of the graph is learned by the GCN.

A GCN is usually composed of a set of graph convolutional layers that are sequentially stacked together. Formally, let $\mathbf{H}^l \in \mathbb{R}^{K \times F^l}$ be the input to a graph convolutional layer, $\mathbf{A}^l \in \mathbb{R}^{K \times K}$ the adjacency matrix, and $\mathbf{W}^l \in \mathbb{R}^{F^l \times F^{l+1}}$ the trainable parameters, the output of the graph convolutional layer is:

$$\mathbf{H}^{l+1} = \sigma(\mathbf{A}^l \mathbf{H}^l \mathbf{W}^l), \tag{1}$$

where $\mathbf{H}^{l+1} \in \mathbb{R}^{K \times F^{l+1}}$, and $\sigma(\cdot)$ is an activation function.

To map the input pose sequence to the target pose sequence, we design one start GCN, one end GCN, and 6 residual GCNs, the architecture of which is shown in Figure 2. The start GCN has 3 graph convolutional layers, projecting the input pose sequence from the space of $\mathbb{R}^{K \times T}$ to $\mathbb{R}^{K \times F}$, with $F = 256$ in this paper. Followed by 6 residual GCNs each contain 6 graph convolutional layers, which accept features in space $\mathbb{R}^{K \times F}$ and outputs features in the same space. Finally, the end GCN, also containing 3 graph convolutional layers, projects the features in space $\mathbb{R}^{K \times F}$ to the target pose sequence in space $\mathbb{R}^{K \times T}$. The whole network learns the residual vector between the input and target pose sequences by adding a skip connection.

Note that the above pose prediction network with basic Graph Convolution Blocks is similar to the method proposed in [34] except for the Discrete Cosine Transform (DCT) and inverse DCT components for data representation transformation. It can achieve similar results as [34] while performing much better than earlier works [35, 23]. However, it can be further improved by taking advantage of the multi-scale properties of human pose [26].

3.2. Multi-scale Residual GCNs

Intuitively, a human pose can be simplified step by step to obtain a set of fine-to-coarse poses. With the increase of the coarse-scale, the motion of the pose becomes more stable, which usually means the pose prediction in this scale

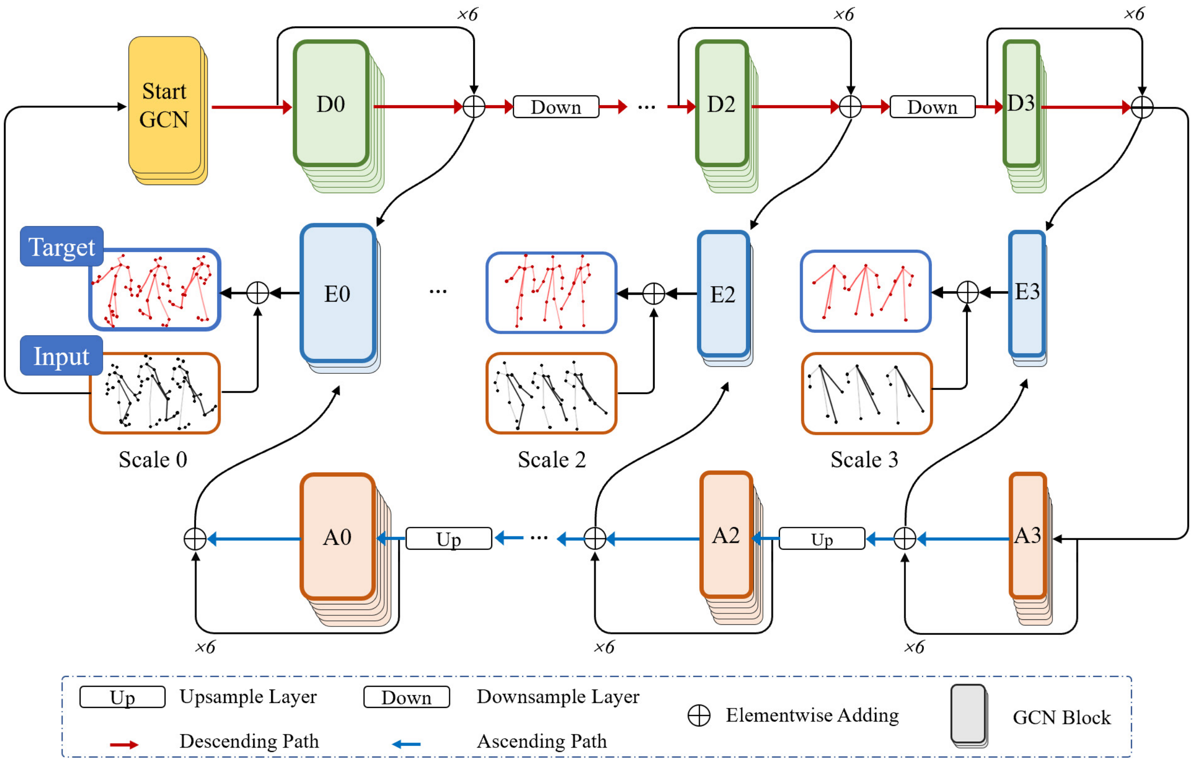

Figure 3. The architecture of the proposed MSR-GCN which consists of one start GCN, four descending GCNs ($D0$, $D1$, $D2$, $D3$), four ascending GCNs ($A0$, $A1$, $A2$, $A3$), and four end GCNs ($E0$, $E1$, $E2$, $E3$). The start GCN only takes the black poses at scale 0 as input. Then descending and ascending GCNs are stacked sequentially to extract features for each scale twice. The combined features at each scale are finally fed into the corresponding end GCN for decoding. Residual connections are added after every end GCN to add the ground truth poses to the output of each GCN, making the network learn residuals rather than the target poses directly.

is easier than a finer scale. This motivates us to propose a Multi-scale Residual Graph Convolution Network (MSR-GCN), in which we perform prediction at the coarsest level firstly, and then go up to higher levels step by step. As shown in Figure 3, our MSR-GCN is composed of four kinds of GCNs: one start GCN, a set of descending and ascending GCN blocks, and a set of end or decoding GCNs.

Before introducing MSR-GCN, let us describe how we abstract a human pose. As shown in the leftmost picture of Figure 1, the finest pose has 22 joints. We abstract the finest pose recursively to obtain 3 poses with 10, 7, and 4 joints respectively. The subplots in the second row of Figure 1 (from left to right) depict how to combine the joints at the finer level, while those in the first row show the obtained poses at the next levels correspondingly. Note that we also try other grouping manners, but find this scheme yields the most stable motion at the coarsest level, corresponding comparison results are shown in Section 4.4.

Start GCN is composed of 3 convolutional layers, mapping the input poses to the feature space. The pose space is $\mathbb{R}^{K\times T}$ as defined above, and the feature space is $\mathbb{R}^{K\times F}$ with F setted as 256. We just utilize the finest-scale pose sequence as the input while the pose sequences at other scales are only used at end GCNs to calculate residuals. This allows us to extract features at different scales automatically from the finest input by using the intermediate supervision over all other scales, which will be described later.

Descending and ascending GCN blocks. Since we have abstracted the human pose in four levels, we use four descending and four ascending GCN blocks, namely $D0, D1, D2, D3$ and $A3, A2, A1, A0$, to extract features at the four scales. Each of these blocks loops a residual GCN 6 times, and each GCN has 6 graph convolutional layers. The eight GCN blocks are sequentially stacked together. Along the whole descending and ascending path, the feature dimension F is always kept as 256, but the pose dimension K changes between adjacent descending or ascending blocks. For example, $D0$ extracts features in space $\mathbb{R}^{K_0\times F}$ with $K_0 = 22 \times 3 = 66$, while $K_1 = 36$, $K_2 = 21$ and $K_3 = 12$ for $D1$, $D2$ and $D3$. We use a downsampling layer to transform the features outputted by $D0$ into the space of $\mathbb{R}^{K_1\times F}$. The descending blocks gradually reduce the pose dimension which is then gradually increased by the ascending blocks with upsampling layers. We concatenate the features extracted by a descending GCN block and the corresponding ascending GCN block together and deliver them to the end GCNs for decoding.

End GCNs are used for decoding the concatenated features extracted by descending and ascending blocks to poses. Like start GCN, an end GCN is also composed of

3 graph convolutional layers. But instead of just one start GCN, we design 4 end GCNs, namely $E0, E1, E2, E3$, to decode combined features at four different scales, respectively. The intermediate supervision used to train the whole network is obtained by computing the L2 distance between the decoded poses and their ground truth at all scales. In fact, it is a commonly adopted strategy in many works [48, 52]. Ablation experiments show that with the intermediate supervision, better prediction accuracy can be obtained, which we conjecture is because it helps extract more representative features in coarser levels and enforce the whole network to learn the prediction from coarse to fine-scale. Finally, the output of “E0” is the predicted target pose sequence.

Residual Connections. Besides the residual connections in descending and ascending GCNs, we add a residual connection after each end GCN, that is to say, we add the input pose sequence (at different scales) to the output of the end GCN. In this way, the MSR-GCN learns the residual vector between the input and ground truth at all levels.

3.3. Implementation Details

We choose Adam as the optimizer with the initial learning rate of 2e-4, which decays by 0.98 every two epochs and train the network on an NVIDIA RTX 3090 GPU card.

4. Experiments

To verify the effectiveness of our MSR-GCN, we run experiments on two standard benchmark motion capture datasets, including Human3.6M (H3.6M) and CMU Mocap dataset. Here we will first introduce the two datasets, the evaluation metric and the baselines we compare with, then present experimental results and ablation analysis.

4.1. Datasets Setup

The **H3.6M** dataset consists of seven subjects S1, S5, S6, S7, S8, S9, and S11, each one contains 15 action categories. We transform the origin data from exponential mapping (expmap) format into the 3D joint coordinate space, downsample the original pose sequence by 2 along the time axis and choose 22 body joints from the origin 32 joints of a single spatial pose. Like [35, 26, 34], we use the data of S5 and S11 as test and validation dataset respectively, and the rest data is used for training. We use four scales in descending and ascending section, which contains 22, 12, 7, and 4 joints respectively.

The **CMU Mocap** dataset is another commonly used dataset for human pose prediction, which includes 8 action categories. A single pose has 38 body joints in the original dataset, among which we choose 25 and abstract to 12, 7, and 4 joints. Other details are set similar to H3.6M.

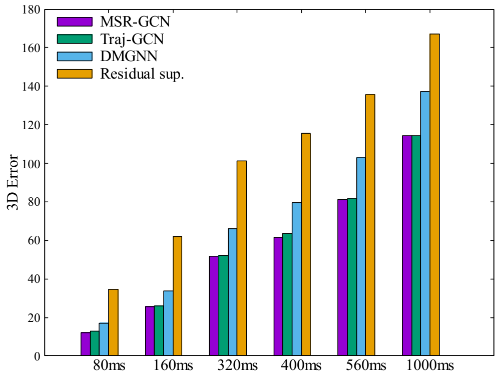

Figure 4. Comparison of average prediction error (APE) over all actions at different forecast time on the H3.6M dataset.

4.2. Comparison Settings

Metrics. Mean Per Joint Position Error (MPJPE) in millimeter is the most widely used evaluation metric. Supposing the predicted pose sequence is $\hat{X}_{1:T}$ and the corresponding ground truth is $X_{1:T}$, then the MPJPE loss is

$$\mathcal{L}_{\text{MPJPE}} = \frac{1}{J \times T} \sum_{t=1}^{T} \sum_{j=1}^{J} \|\hat{p}_{j,t} - p_{j,t}\|^2, \tag{2}$$

where $\hat{p}_{j,t} \in \mathbb{R}^3$ represents the predicted j-th joint position in frame t, and $p_{j,t}$ is the corresponding ground truth.

Baselines. We compare our approach with three state-of-the-art baselines, *i.e.*, denoted as Residual sup. [35], DMGNN [26], and Traj-GCN [34], respectively. The [35] is based on RNN, and the rest two are based on GCNs. Specifically, [26] builds a dynamic multi-scale graph convolution neural network, and [34] transforms the original data from 3D coordinate space to frequency space.

Random test batch *vs*. full test set. All the compared three works [35, 26, 34] evaluate their methods on just one randomly selected single batch data of size 8 for each action category. We argue that such little test data leads to high variance which has also been questioned in [36]. To alleviate this problem, we modify their published codes and totally retrain the networks to use the whole test dataset in 3D coordinate space to evaluate the MPJPE, at the same time experimental results with the exact same evaluation manner from prior work can also be found in Section 4.3.

Unifying input and output length. Methods of [35, 26] require 50 historical observed poses to predict 25 future poses, while [34] predicts 25 future poses by just 10 poses. All the experiments in this paper follow the way of [34].

4.3. Results

To validate the prediction performance of our MSR-GCN, we show qualitative and quantitative comparisons for 400ms short-term (*i.e.*, 10 frames) and 1000ms long-term (*i.e.*, 25 frames) results on H3.6M and CMU Mocap.

Results on H3.6M. The quantitative comparison for both short-term and long-term prediction results are pre-

scenarios	walking				eating				smoking				discussion			
millisecond (ms)	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Residual sup. [35]	29.36	50.82	76.03	81.51	16.84	30.60	56.92	68.65	22.96	42.64	70.14	82.68	32.94	61.18	90.92	96.19
DMGNN [26]	17.32	30.67	54.56	65.20	10.96	21.39	36.18	43.88	8.97	17.62	32.05	40.30	17.33	34.78	61.03	69.80
Traj-GCN [34]	12.29	23.03	39.77	46.12	**8.36**	**16.90**	33.19	40.70	**7.94**	**16.24**	31.90	38.90	12.50	27.40	58.51	71.68
MSR-GCN	**12.16**	**22.65**	**38.64**	**45.24**	8.39	17.05	**33.03**	**40.43**	8.02	16.27	**31.32**	**38.15**	**11.98**	**26.76**	**57.08**	**69.74**
scenarios	directions				greeting				phoning				posing			
millisecond (ms)	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Residual sup. [35]	35.36	57.27	76.30	87.67	34.46	63.36	124.60	142.50	37.96	69.32	115.00	126.73	36.10	69.12	130.46	157.08
DMGNN [26]	13.14	24.62	64.68	81.86	23.30	50.32	107.30	132.10	12.47	25.77	48.08	58.29	15.27	29.27	71.54	96.65
Traj-GCN [34]	8.97	19.87	43.35	**53.74**	18.65	38.68	77.74	93.39	10.24	21.02	42.54	52.30	13.66	29.89	**66.62**	**84.05**
MSR-GCN	**8.61**	**19.65**	**43.28**	53.82	**16.48**	**36.95**	**77.32**	**93.38**	**10.10**	**20.74**	**41.51**	**51.26**	**12.79**	**29.38**	66.95	85.01
scenarios	purchases				sitting				sittingdown				takingphoto			
millisecond (ms)	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Residual sup. [35]	36.33	60.30	86.53	95.92	42.55	81.40	134.70	151.78	47.28	85.95	145.75	168.86	26.10	47.61	81.40	94.73
DMGNN [26]	21.35	38.71	75.67	92.74	11.92	25.11	44.59	50.20	14.95	32.88	77.06	93.00	13.61	28.95	45.99	58.76
Traj-GCN [34]	15.60	32.78	**65.72**	**79.25**	10.62	**21.90**	46.33	57.91	16.14	**31.12**	**61.47**	**75.46**	**9.88**	**20.89**	44.95	56.58
MSR-GCN	**14.75**	**32.39**	66.13	79.64	**10.53**	21.99	**46.26**	**57.80**	**16.10**	31.63	62.45	76.84	9.89	21.01	**44.56**	**56.30**
scenarios	waiting				walkingdog				walkingtogether				Average			
millisecond (ms)	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Residual sup. [35]	30.62	57.82	106.22	121.45	64.18	102.10	141.07	164.35	26.79	50.07	80.16	92.23	34.66	61.97	101.08	115.49
DMGNN [26]	12.20	24.17	59.62	77.54	47.09	93.33	160.13	171.20	14.34	26.67	50.08	63.22	16.95	33.62	65.90	79.65
Traj-GCN [34]	11.43	23.99	50.06	61.48	23.39	46.17	83.47	95.96	10.47	21.04	38.47	45.19	12.68	26.06	52.27	63.51
MSR-GCN	**10.68**	**23.06**	**48.25**	**59.23**	**20.65**	**42.88**	**80.35**	**93.31**	**10.56**	**20.92**	**37.40**	**43.85**	**12.11**	**25.56**	**51.64**	**62.93**

Table 1. Comparisons of 3D error for short-term prediction on the H3.6M dataset. The best results are highlighted in bold.

scenarios	walking		Eating		Smoking		Discussion		Directions		average	
millisecond (ms)	560	1000	560	1000	560	1000	560	1000	560	1000	560	1000
Residual sup.[35]	81.73	100.68	79.87	100.20	94.83	137.44	121.30	161.70	110.05	152.48	97.56	130.50
DMGNN [26]	73.36	95.82	58.11	86.66	50.85	72.15	**81.90**	138.32	110.06	115.75	74.85	101.74
Traj-GCN [34]	54.05	**59.75**	53.39	77.75	50.74	72.62	91.61	121.53	**71.01**	101.79	64.16	86.69
MSR-GCN	**52.72**	63.04	**52.54**	**77.11**	**49.45**	**71.64**	88.59	**117.59**	71.18	**100.59**	**62.89**	**86.00**

Table 2. Comparisons of 3D error for long-term prediction on five selected action scenarios of the H3.6M dataset and their averages. The best results are highlighted in bold.

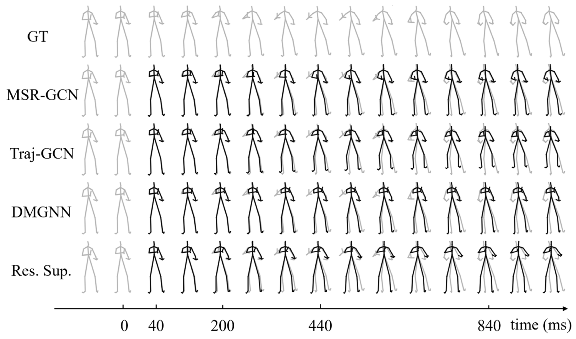

Figure 5. Visualization of predicted poses of different methods on a sample of the H3.6M dataset. With the increase of forecast time, our result is better than results of other methods.

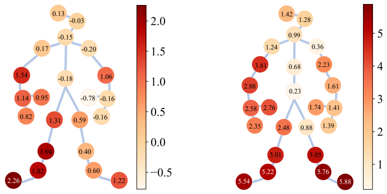

Figure 6. Average performance gain over Traj-GCN [34] of joints on H36M (left) and CMU (right).

sented in Table 1 and Table 2 respectively. Our proposed MSR-GCN outperforms the compared methods on all 15 action scenarios of Human3.6M in most cases. Apparently, the performance gap between the RNN model Residual sup. [35] and other GCN models is very large. Although both are GCN-based methods, our MSR-GCN is much better than DMGNN. This is partly because the GCNs of DMGNN are constrained to the natural connections of human joints while our GCNs can learn the relation between any pair of joints. Another reason may due to that DMGNN utilizes a GRU-based RNN model to predict future poses one by one, but our method directly outputs the whole pose sequence. The results of Traj-GCN [34] is quite good, but our MSR-GCN steadily outperforms it on account of the newly introduced multi-scale architecture built up in a descending-ascending manner and the intermediate losses applied on lower-level scales.

For more intuitive comparison, we plot the average prediction error (APE) over all kinds of actions of different methods at different forecast time in Figure 4, which clearly shows our MSR-GCN outperforms the compared three methods. Figure 5 shows an example of the predicted poses of different methods. In this example, with the increase of the forecast time, the result of MSR-GCN is better than those of others.

Results on CMU Mocap. We also conduct quantitative comparisons for short-term and long-term pose prediction on the CMU Mocap dataset, as shown in Table 3 and Table 4. The APE at each forecast time is also computed for the short-term prediction. Our method gets the best average performance for all short-term predictions. For long-term prediction, *i.e.*, predicting the frame after 1000ms, our method achieves the best results on four kinds of actions. For other actions, the prediction errors of our method are always the second best and are very close to the best ones.

scenarios	basketball				basketball signal				directing traffic				jumping			
millisecond (ms)	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Residual sup. [35]	15.45	26.88	43.51	49.23	20.17	32.98	42.75	44.65	20.52	40.58	75.38	90.36	26.85	48.07	93.50	108.90
DMGNN [26]	15.57	28.72	59.01	73.05	5.03	9.28	20.21	26.23	10.21	20.90	41.55	52.28	31.97	54.32	96.66	119.92
Traj-GCN [34]	11.68	21.26	40.99	50.78	3.33	6.25	13.58	17.98	6.92	13.69	30.30	39.97	17.18	32.37	60.12	72.55
MSR-GCN	**10.28**	**18.94**	**37.68**	**47.03**	**3.03**	**5.68**	**12.35**	**16.26**	**5.92**	**12.09**	**28.36**	**38.04**	**14.99**	**28.66**	**55.86**	**69.05**
scenarios	running				soccer				walking				washwindow			
millisecond (ms)	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Residual sup. [35]	25.76	48.91	88.19	100.80	17.75	31.30	52.55	61.40	44.35	76.66	126.83	151.43	22.84	44.71	86.78	104.68
DMGNN [26]	17.42	26.82	38.27	40.08	14.86	25.29	52.21	65.42	9.57	15.53	26.03	30.37	7.93	14.68	33.34	44.24
Traj-GCN [34]	14.53	24.20	37.44	41.10	13.33	24.00	43.77	53.20	6.62	10.74	**17.40**	**20.35**	5.96	11.62	**24.77**	**31.63**
MSR-GCN	**12.84**	**20.42**	**30.58**	**34.42**	**10.92**	**19.50**	**37.05**	**46.38**	**6.31**	**10.30**	17.64	21.12	**5.49**	**11.07**	25.05	32.51

Table 3. Short-term prediction error of 3D joint positions on the CMU Mocap dataset. The best results are highlighted in bold.

scenarios	basket	bas_sig	dir_tra	jumping
Residual sup.[35]	**72.83**	60.57	153.12	162.84
DMGNN [26]	138.62	52.04	111.23	224.63
Traj-GCN [34]	97.99	54.00	114.16	127.41
MSR-GCN	86.96	**47.91**	**111.04**	**124.79**
scenarios	running	soccer	walking	washwin
Residual sup.[35]	158.19	107.37	194.33	202.73
DMGNN [26]	**46.40**	111.90	67.01	82.84
Traj-GCN [34]	51.73	108.26	**34.41**	**66.95**
MSR-GCN	48.03	**99.32**	39.70	71.30

Table 4. Long-term prediction of 1000 ms on the CMU Mocap dataset. The best results are highlighted in bold.

	H3.6M		CMU	
	short-term	long-term	short-term	long-term
Traj-GCN [34]	37.35	59.02	29.13	45.06
Ours	**36.32**	**57.84**	**24.84**	**41.48**

Table 5. APE using the same evaluation setting (*i.e.*, test on 8 random samples per action).

Time (ms)	80	160	320	40	560	1000
Human3.6M	-0.23	-0.32	**0.28**	**0.95**	**2.92**	**5.80**
CMU	**1.14**	**2.16**	**3.45**	**3.62**	**3.63**	**4.25**

Table 6. Average performance gain over [34] at different timesteps.

Given that the performance gain of our MSR-GCN over Traj-GCN [34] is not that large, we have also trained our method and [34] five times with random seeds and tested the performance on Human3.6M and CMU. The specific prediction errors of our method are 58.37±0.43 and 37.52±0.48 on the two datasets. In comparison, [34] reports higher predictor errors and larger variances than our method, which are 59.93±0.91 on Human3.6M and 40.56±0.50 on CMU.

In-depth analysis and reasoning of performance gain. We have examined our performance gain over Traj-GCN [34] for each joint, and found that larger performance gains are achieved for joints of limbs, as shown in Figure 6 where deeper red color means higher performance gain. Since joints on the limbs usually have higher motion frequency, the figure also indicates our method can better handle high-frequency motions. Moreover, as verified in Table 6, our method works better than Traj-GCN [34] in handling challenging long-term motions.

Same evaluation Results. The APE using the same evaluation settings (*i.e.*, test on randomly selected 8 samples per action) are shown in Table 5 which proves the superiority of our MSR-GCN fairly and clearly.

4.4. Ablation Study

The influences of several key elements of our proposed model, such as the scale level number, the intermediate supervision losses, the residual GCNs, and the multi-scale grouping manner are investigated on the CMU Mocap dataset to provide a deeper understanding of our approach. Specifically, we modify MSR-GCN to obtain five ablation variants of it: (1) MSR-GCN w/o inter-loss: the MSR-GCN without intermediate supervision losses, (2) MSR-GCN-3L: the MSR-GCN with three pose scales (note that the original MSR-GCN has four scales), (3) and (4) MSR-GCN-2L, and MSR-GCN-1L with two scales and one scale respectively, (5) MSR-FCL: replace the residual GCNs by residual fully connected layers.

Effects of multi-scale architecture. To study the multi-scale mechanism of the proposed architecture, we conduct experiments on three-scale, two-scale and one-scale structures besides the proposed four-scale MSR-GCN. The comparison results are shown in Table 7. Please see the rows corresponding to MSR-GCN, MSR-GCN-3L, MSR-GCN-2L, and MSR-GCN-1L. In most cases, MSR-GCN is the best, followed by MSR-GCN-3L, MSR-GCN-2L and MSR-GCN-1L. As an example, for the action of running, the prediction error of the four variants at time 320ms are 30.58, 35.87, 38.95, and 39.06, respectively. Obviously, the mentioned four variants MSR-GCN-1L/2L/3L & MSR-GCN indeed gradually increases the accuracy compared with [34], which convincingly demonstrate the effectiveness of our multi-scale architecture. We have also compared our multi-scale architecture with the DCT architecture of [34] on the CMU dataset. Results show the performance gain led by DCT is 0.55 (APE 39.75 of [34] w/ DCT *vs*. 40.30 of [34] w/o DCT), while that of our multi-scale strategy is 3.15 (APE 37.28 of MSR-GCN *vs*. 40.43 of MSR-GCN-1L), which manifests that directly computing global residuals between padded input poses and the target poses without translating to DCT coefficients is effective enough and computationally more efficient.

Effects of intermediate supervision. The effects of intermediate losses on the lower-level scales are analyzed by removing the "End GCN" of the second, the third, and the fourth scale from MSR-GCN. Please see the two rows corresponding MSR-GCN and MSR-GCN w/o inter-loss in Table 7 to compare the two variants. In most cases, MSR-GCN is better than MSR-GCN w/o inter-loss, which demonstrates the necessity of intermediate supervision. Although some exceptions happen on "walking" and "jump-

								running					soccer				
	s1	s2	s3	s4	inter-loss	GCB	FCL	80	160	320	400	1000	80	160	320	400	1000
MSR-GCN	✓	✓	✓	✓	✓	✓		**12.84**	**20.42**	**30.58**	**34.42**	**48.03**	**10.92**	**19.50**	**37.05**	**46.38**	**99.32**
MSR-GCN w/o inter-loss	✓	✓	✓	✓		✓		13.20	21.20	32.69	36.02	51.65	11.03	19.81	38.93	48.84	101.36
MSR-GCN-3L	✓	✓	✓		✓	✓		13.60	22.79	35.87	39.58	49.60	11.02	19.84	38.49	48.26	107.17
MSR-GCN-2L	✓	✓			✓	✓		14.30	23.37	38.95	45.11	73.26	10.93	19.62	38.44	48.30	106.35
MSR-GCN-1L	✓				✓	✓		14.24	24.21	39.06	43.60	74.52	11.55	21.37	43.26	55.00	123.69
MSR-FCL	✓	✓	✓	✓	✓		✓	13.33	24.29	43.58	50.01	61.90	12.16	22.83	46.49	59.04	132.47

								walking					jumping				
	s1	s2	s3	s4	inter-loss	GCB	FCL	80	160	320	400	1000	80	160	320	400	1000
MSR-GCN	✓	✓	✓	✓	✓	✓		**6.31**	**10.30**	17.64	21.12	39.70	14.99	28.66	**55.86**	**69.05**	**124.79**
MSR-GCN w/o inter-loss	✓	✓	✓	✓		✓		6.36	10.33	**17.05**	**20.04**	**34.67**	**14.65**	**28.22**	56.43	70.07	125.69
MSR-GCN-3L	✓	✓	✓		✓	✓		6.62	10.91	18.10	21.19	42.72	14.98	28.89	57.69	71.60	128.62
MSR-GCN-2L	✓	✓			✓	✓		7.87	13.41	23.16	27.63	52.31	15.21	29.67	59.85	74.31	128.10
MSR-GCN-1L	✓				✓	✓		6.73	11.09	17.94	20.95	37.21	15.49	29.73	58.94	73.10	131.72
MSR-FCL	✓	✓	✓	✓	✓		✓	7.19	12.58	23.15	28.00	52.77	15.14	29.89	61.31	76.49	139.01

Table 7. Influence of the scale level number, intermediate loss, graph convolution blocks and fully-connected blocks on 8 action scenarios of the CMU Mocap dataset. Note that, on average, all the designs of our model contribute to its accuracy.

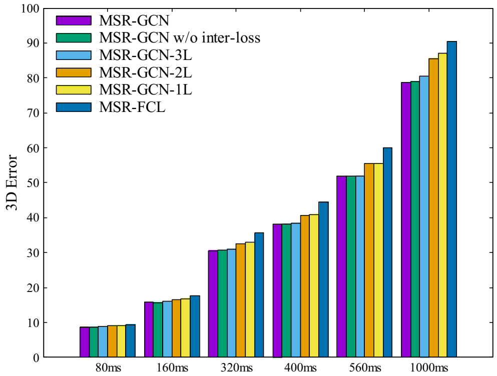

Figure 7. Comparison of APE over all kinds of actions of different ablation variants at different forecast time.

ing", the difference in results between the two variants are very small.

Effects of GCNs. We replace all the GCBs with plain networks just comprising fully connected layers (FCL) and compare the differences to analyze the effects of the GCNs. Please see the rows corresponding to MSR-GCN and MSR-FCL of Table 7. The experimental results show that MSR-GCN is better than MSR-FCL by a large margin. This strongly validates the importance of GCN for high-accuracy pose prediction.

Effects of global residuals The APEs on the CMU dataset show the global residual (GR) leads to noticeable performance gains for both [34] (APE 39.75 w/ GR *vs*. 49.82 w/o GR) and our MSR-GCN (APE 37.28 *vs*. 46.92). Nevertheless, ours without GR (APE 46.92) still clearly outperforms other baselines without GR ([34] w/o residual (APE 49.82) and DMGNN (APE 53.05)), that is because the GR can usually provide good initial guesses for the target poses.

Effects of multi-scale grouping manners. We compare the APE on the CMU dataset of our method (APE 37.28) and other different grouping strategies, including 25-10-5-3 (specified)(APE 40.99) and three random groupings of the default 25-12-7-4 (APE 41.15, 45.77 and 47.04).

More visualizations are shown in Figure 7 and Figure 8. In Figure 7, we show the APE over all kinds of actions of different ablation variants at different forecast time. As can be seen, in this figure, MSR-GCN is always better than its variants, without any exception. In Figure 8, we show an example of the predicted poses by different ablation variants, which clearly demonstrates that MSR-GCN is much better than MSR-GCN-2L, MSR-GCN-1L, and MSR-FCL, verifying the necessity of both the building blocks of GCN and the multi-scale architecture.

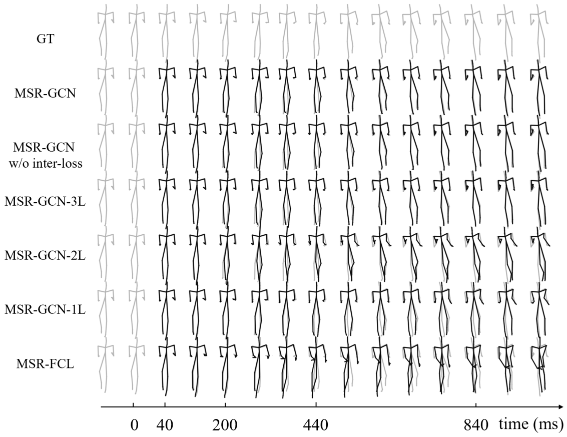

Figure 8. Visualization of predicted pose by different ablation variants on a sample of the CMU Mocap dataset.

5. Conclusion

In this paper, we build a multi-scale residual graph convolution network to effectively predict future human motion sequences from the observed historical ones. Losses are added to all scales to provide intermediate supervision. We use a shorter observed historical pose sequence of 10 frames as inputs to predict future 25 frames. We argue that the past testing method which uses 8 randomly selected samples of each action scenario would produce a high variance and the model should better be evaluated on the whole test dataset. Our approach outperforms current state-of-the-art methods on two standard benchmark datasets. We will further explore the multiple-scale organization manners in the future.

References

[1] Emre Aksan, Peng Cao, Manuel Kaufmann, and Otmar Hilliges. A spatio-temporal transformer for 3d human motion prediction. *arXiv e-prints*, pages arXiv–2004, 2020. 2

[2] Amal Fahad Al-aqel and Murtaza Ali Khan. Attention mechanism for human motion prediction. In *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)*, pages 1–6. IEEE, 2020. 1, 2

[3] Matthew Brand and Aaron Hertzmann. Style machines. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 183–192, 2000. 1

[4] Yujun Cai, Lin Huang, Yiwei Wang, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Xu Yang, Yiheng Zhu, Xiaohui Shen, et al. Learning progressive joint propagation for human motion prediction. In *European Conference on Computer Vision*, pages 226–242. Springer, 2020. 2

[5] Wensong Chan, Zhiqiang Tian, and Yang Wu. Gas-gcn: Gated action-specific graph convolutional networks for skeleton-based action recognition. *Sensors*, 20(12):3499, 2020. 1, 2

[6] Qiongjie Cui, Huaijiang Sun, Yue Kong, Xiaoqian Zhang, and Yanmeng Li. Efficient human motion prediction using temporal convolutional generative adversarial network. *Information Sciences*, 545:427–447, 2021. 1, 2

[7] Qiongjie Cui, Huaijiang Sun, and Fei Yang. Learning dynamic relationships for 3d human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6519–6527, 2020. 2, 3

[8] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015. 1, 2

[9] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José MF Moura. Adversarial geometry-aware human motion prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 786–803, 2018. 1, 2

[10] Xiao Guo and Jongmoo Choi. Human motion prediction via learning local structure representations and temporal dependencies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2580–2587, 2019. 1, 2

[11] Alejandro Hernandez, Jurgen Gall, and Francesc Moreno-Noguer. Human motion prediction via spatio-temporal inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7134–7143, 2019. 1, 2

[12] Danfeng Hong, Lianru Gao, Jing Yao, Bing Zhang, Antonio Plaza, and Jocelyn Chanussot. Graph convolutional networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 2020. 2

[13] Tao Hu, Chengjiang Long, and Chunxia Xiao. Crd-cgan: Category-consistent and relativistic constraints for diverse text-to-image generation. *arXiv preprint arXiv:2107.13516*, 2021. 2

[14] Tao Hu, Chengjiang Long, and Chunxia Xiao. A novel visual representation on text using diverse conditional gan for visual recognition. *IEEE Transactions on Image Processing*, 30:3499–3512, 2021. 2

[15] Gang Hua, Chengjiang Long, Ming Yang, and Yan Gao. Collaborative active learning of a kernel machine ensemble for recognition. In *IEEE International Conference on Computer Vision*. IEEE, 2013. 2

[16] Gang Hua, Chengjiang Long, Ming Yang, and Yan Gao. Collaborative active visual recognition from crowds: A distributed ensemble approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):582–594, 2018. 2

[17] Ashraful Islam, Chengjiang Long, Arslan Basharat, and Anthony Hoogs. Doa-gan: Dual-order attentive generative adversarial network for image copy-move forgery detection and localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[18] Ashraful Islam, Chengjiang Long, and Richard Radke. A hybrid attention mechanism for weakly-supervised temporal action localization. In *AAAI Conference on Artificial Intelligence*, 2021. 2

[19] Qiuhong Ke, Mohammed Bennamoun, Hossein Rahmani, Senjian An, Ferdous Sohel, and Farid Boussaid. Learning latent global network for skeleton-based action prediction. *IEEE Transactions on Image Processing*, 29:959–970, 2019. 1, 2

[20] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*, 2016. 3

[21] Jogendra Nath Kundu, Maharshi Gor, and R Venkatesh Babu. Bihmp-gan: Bidirectional 3d human motion prediction gan. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8553–8560, 2019. 1, 2

[22] Andreas M Lehrmann, Peter V Gehler, and Sebastian Nowozin. Efficient nonlinear markov models for human motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1314–1321, 2014. 1

[23] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional sequence to sequence model for human dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5226–5234, 2018. 3

[24] Fanjia Li, Aichun Zhu, Yonggang Xu, Ran Cui, and Gang Hua. Multi-stream and enhanced spatial-temporal graph convolution network for skeleton-based action recognition. *IEEE Access*, 8:97757–97770, 2020. 2

[25] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3595–3603, 2019. 2

[26] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 214–223, 2020. 2, 3, 5, 6, 7

[27] Xiaoli Liu, Jianqin Yin, Jin Liu, Pengxiang Ding, Jun Liu, and Huaping Liub. Trajectorycnn: a new spatio-temporal feature learning network for human motion prediction. *IEEE*

Transactions on Circuits and Systems for Video Technology, 2020. 1, 2

[28] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 143–152, 2020. 2

[29] Chengjiang Long, Roddy Collins, Eran Swears, and Anthony Hoogs. Deep neural networks in fully connected crf for image labeling with social network metadata. In *2019 IEEE Winter Conference on Applications of Computer Vision*, 2019. 2

[30] Chengjiang Long and Gang Hua. Multi-class multi-annotator active learning with robust gaussian process for visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 2

[31] Chengjiang Long and Gang Hua. Correlational gaussian processes for cross-domain visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[32] Chengjiang Long, Gang Hua, and Ashish Kapoor. A joint gaussian process model for active visual recognition with expertise estimation in crowdsourcing. *International Journal of Computer Vision*, 116(2):136–160, 2016. 2

[33] Chengjiang Long, Xiaoyu Wang, Gang Hua, Ming Yang, and Yuanqing Lin. Accurate object detection with location relaxation and regionlets re-localization. In *The 12th Asian Conference on Computer Vision*, 2014. 2

[34] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9489–9497, 2019. 2, 3, 5, 6, 7, 8

[35] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2891–2900, 2017. 1, 2, 3, 5, 6, 7

[36] Dario Pavllo, Christoph Feichtenhofer, Michael Auli, and David Grangier. Modeling human motion with quaternion-based neural networks. *International Journal of Computer Vision*, pages 1–18, 2019. 5

[37] Vladimir Pavlovic, James M Rehg, and John MacCormick. Learning switching linear models of human motion. In *NIPS*, volume 2, page 4, 2000. 1

[38] Hai-Feng Sang, Zi-Zhen Chen, and Da-Kuo He. Human motion prediction based on attention mechanism. *Multimedia Tools and Applications*, 79(9):5529–5544, 2020. 1, 2

[39] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Mo Zhou, Zhenxing Niu, and Gang Hua. Sgcn: Sparse graph convolution for pedestrian trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[40] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[41] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1227–1236, 2019. 2

[42] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. 1, 2

[43] Yongyi Tang, Lin Ma, Wei Liu, and Weishi Zheng. Long-term human motion prediction by modeling motion context and enhancing motion dynamic. *arXiv preprint arXiv:1805.02513*, 2018. 1, 2

[44] Graham W Taylor, Geoffrey E Hinton, and Sam T Roweis. Modeling human motion using binary latent variables. In *Advances in neural information processing systems*, pages 1345–1352. Citeseer, 2007. 1

[45] Dong Wang, Yuan Yuan, and Qi Wang. Early action prediction with generative adversarial networks. *IEEE Access*, 7:35795–35804, 2019. 1, 2

[46] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models. In *NIPS*, volume 18, page 3. Citeseer, 2005. 1

[47] Yongxin Wang, Kris Kitani, and Xinshuo Weng. Joint object detection and multi-object tracking with graph neural networks. *arXiv preprint arXiv:2006.13164*, 2020. 2

[48] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016. 5

[49] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 2

[50] H. Yang, C. Yuan, L. Zhang, Y. Sun, W. Hu, and S. J. Maybank. Sta-cnn: Convolutional spatial-temporal attention learning for action recognition. *IEEE Transactions on Image Processing*, 29:5783–5793, 2020. 1, 2

[51] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7094–7103, 2019. 2

[52] Wenxiao Zhang, Chengjiang Long, Qingan Yan, Alix LH Chow, and Chunxia Xiao. Multi-stage point completion network with critical set supervision. *Computer Aided Geometric Design*, 82:101925, 2020. 5

[53] Xikun Zhang, Chang Xu, and Dacheng Tao. Context aware graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14333–14342, 2020. 2

[54] Tianhang Zheng, S. Liu, Changyou Chen, Junsong Yuan, B. Li, and K. Ren. Towards understanding the adversarial vulnerability of skeleton-based action recognition. *ArXiv*, abs/2005.07151, 2020. 1, 2