PREDICTING HUMAN MOTION USING KEY SUBSEQUENCES

Menghao Li[1], *Mingtao Pei*[1], *Wei Liang*[1,2]

[1]Lab of Intelligent Info. Technology, Beijing Institute of Technology, Beijing, China
[2]Yangtze Delta Region Academy of Beijing Institute of Technology, Jiaxing, China

ABSTRACT

Human motion prediction is an important task in computer vision, and has a wide range of applications, such as autonomous driving and human-robot interaction. Usually, human motion tends to repeat itself and follows patterns that are well-represented by a few short key subsequences. Based on the above observations, we propose an attention-based feed-forward network, which is explicitly guided by the key subsequences, for human motion prediction. Specifically, we obtain the key subsequences by clustering, extract motion attention by the similarity between the observed poses and the motion context of corresponding key subsequences, and aggregate the relevant key subsequences by a graph convolutional network to predict human motion. Experimental results on public human motion datasets show that our method achieves better performance over state-of-the-art methods in motion prediction.

Index Terms— Human motion prediction, Clustering, Attention

1. INTRODUCTION

Human motion prediction is to forecast future human poses based on the observed pose data, and has a wide range of applications such as automated driving [1], human-robot interaction [2] and human motion generation [3–5].

Traditional data-driven methods, such as Hidden Markov Model [6] and Gaussian Process Dynamical Models [7], achieve good performance for simple motion categories, such as walking and golf swing. With the development of deep learning and the appearance of large datasets, many deep learning based methods are proposed and achieve promising prediction performance on complicated motions [8–11]. Especially, recurrent neural networks (RNNs), which can handle the sequential data with variable lengths, have shown their strength in capturing temporal dependencies. For instance, Fragkiadaki et.al [8] propose an Encoder-Recurrent-Decoder (ERD) model which incorporates a nonlinear encoder and decoder to process motion dynamics before and after recurrent layers. In [9], the Structural-RNN transforms spatio-temporal graphs to a feedforward mixture of RNNs for motion history encoding. However, RNNs suffer from long-term horizon prediction and tend to generate cumulation errors throughout the predicted process [12]. To address these issues, Li et.al [12] propose to use convolutional networks to respectively model the spatial and temporal dependencies by stacking convolution layers.

Mao et.al [13] argue that humans tend to repeat their motion and introduce an attention model to discover subsequences in the historical motion segment that are similar to the current subsequence. However, the similarity is computed between current subsequence and all the subsequences in the historical segment. Taking efficiency into consideration, only a short historical segment can be used, and the subsequences in the far past will be ignored. In this paper, we obtain key subsequences from the whole motion history by clustering to employ the information contained in the whole motion history, extract motion attention by the similarity between the observed poses and the motion context of corresponding key subsequences, and aggregate the relevant key subsequences by a graph convolutional network to predict human motion.

Most of current methods [13, 14] use the Mean Per Joint Position Error (MPJPE) as the loss function to quantize the difference between the predicted poses and the ground truth. However, the MPJPE only considers the difference between the individual joint pair, and ignores the correlation across different joints. Therefore, we introduce a novel loss function called joint crossover loss which considers the correlation across different joints and can help to improve the prediction performance.

Our contributions are summarized as follows. (i) We obtain key subsequences from the whole motion history by clustering to fully employ the information contained in the whole motion history. (ii) We introduce the joint crossover loss to learn the dependencies across different joints. (iii) We conduct extensive experiments on two public datasets. Experimental results show that our method achieves better performance than state-of-the-art methods for human motion prediction, which experimentally proves the effectiveness of our method.

2. THE PROPOSED METHOD

In human motion prediction, we forecast future pose sequence based on the observed pose sequence. Here, the observed

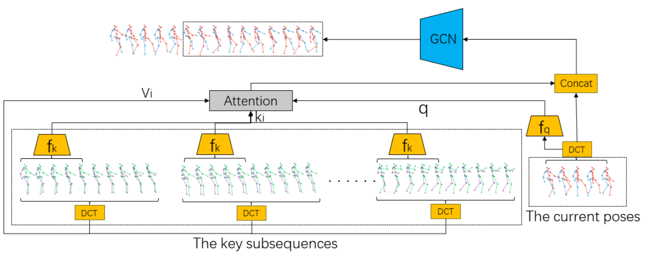

Fig. 1. Overview of our approach. The key subsequences are shown as green and purple skeletons while the current and predicted poses are drawn in red and blue. The current N poses are used as query. For each key subsequence, an attention score is computed to weight the DCT coefficients (values) of the corresponding future poses. We concatenate the weighted sum of the DCT coefficients of the current poses, and pass the values to the GCN based framework provided by HisRep [13] to predict the future motion.

pose sequence is defined by $X_{1:N} = [x_1, x_2, x_3, \cdot\cdot\cdot, x_N]$ consisting of N consecutive poses, where $x_i \in \mathbb{R}^K$, and K is the number of parameters describing each pose. Then our goal is to predict the poses $X_{N+1:N+T}$ for the future T time steps. To this end, we propose a key subsequence attention model to predict future motion estimate by aggregating the temporal information from the whole motion histories. Then we combine the predicted estimate with the observed motion, and employ graph convolutional networks (GCNs) to capture the spatio-temporal dependencies. Note that we adopt a 3D-joint position representation of human motion which has been gaining popularity over Euler angles, due to the latter one suffering from ambiguities.

2.1. Key Subsequences

For a specific human motion, a key subsequence S_i corresponds to a subsequence with length of N. S_i is defined as a key-value pair (k_i, V_i), where k_i is the learned representation of the key subsequence and V_i is the corresponding learned future motion representation, $i \in [1, C]$, and C is the total number of key subsequences. Here, we use the pretrained two-layer convolutional network [13] to obtain the learned representation for each k_i with the dimension d.

For each motion, all the consecutive subsequences with length $N+T$ are extracted from the training dataset. The first part of each subsequence with length N is used to obtain the learned representation k_i, while the whole subsequence with length $N+T$ is used to obtain the corresponding learned future motion representation V_i. Then these subsequences are then clustered into M clusters via k-means, with the learned representations k_i as keywords for clustering. Finally, the central pair for each cluster is used as key subsequences.

2.2. Motion Attention Module

Following the work of [13], the attention module is described as a mapping from a query and a set of key-value pairs to an output. Here, the query corresponds to a learned representation of the current observed sequence, namely $X_{1:N}$, and the key-value pairs correspond to the key subsequences. The output of attention module is the aggregation of the future motion representations based on the motion similarity between the current observed sequence and the key subsequences. Assume that the subsequence corresponding to the i-th key subsequence S_i is represented as $X^i_{1:N}$. As is discussed in [13], the resulting values are mapped to trajectory space using the DCT on the temporal dimension. Specifically, the final values become the DCT coefficients V_i of the corresponding future motion poses $X^i_{1:N+T}$, where $V_i \in \mathbb{R}^{(N+T)\times K}$. In practice, the DCT can provide a more compact representation and captures the smoothness of human motion by discarding some high frequencies.

As shown in Fig. 1, we use attention mechanism to compute the correlation scores between the query and keys, which then act as attentive weights to aggregate the corresponding values. To this end, we adopt two functions $f_q : \mathbb{R}^{N\times K} \longrightarrow \mathbb{R}^d$ and $f_k : \mathbb{R}^{N\times K} \longrightarrow \mathbb{R}^d$ modeled with the pretrained convolutional network:

$$q = f_q(X_{1:N}), k_i = f_k(X^i_{1:N}), \tag{1}$$

where $q, k_i \in \mathbb{R}^d$ and i $\in \{1, 2, \cdot\cdot\cdot, C\}$. For each key, the correlation score is computed as

$$\alpha_i = \frac{qk_i^T}{\sum_{i=1}^{C} qk_i^T}. \tag{2}$$

With the computed correlation scores, the output of the attention module is thus computed by:

$$U = \sum_{i=1}^{C} \alpha_i V_i. \tag{3}$$

2.3. GCN Based Motion Prediction Model

We adopt the state-of-the-art motion prediction model of [13]. That is, we use DCT-based representation to encode the temporal information, and use GCNs to encode the spatial structure of human poses.

Given the current observed poses $X_{1:N}$, the last pose X_N is replicated T times to generate a sequence of length N+T. Then the DCT coefficients D of this sequence and the key subsequence values U are computed. Finally, the DCT coefficients of the future poses $X_{1:N+T}$ are predicted via the GCN model.

To learn the dependencies among joints, we assume the human body is a fully-connected graph with K nodes. A graph convolutional layer p then takes as input a matrix $H^{(p)} \in$

$\mathbb{R}^{F\times K}$, where F is the dimension of features output by the previous layer. For example, for the first layer, the input is a matrix which caoncatenates D and U. Given this information and the trainable weights $W^{(p)} \in \mathbb{R}^{F\times \hat{F}}$, a graph convolutional layer outputs a matrix:

$$H^{(p+1)} = \sigma(A^{(p)}H^{(p)}W^{(p)}), \tag{4}$$

where $A^{(p)} \in \mathbb{R}^{K\times K}$ is the trainable weighted adjacency matrix for layer p and $\sigma(\cdot)$ is the activation function.

Multiple such layers are stacked as the final motion prediction model, which outputs the future poses using DCT coefficients. Given such coefficients, the original pose representation can be obtained via the Inverse-DCT (IDCT). More details can be found in [13].

2.4. Training

We adopt the Mean Per Joint Position Error (MPJPE) to quantize the difference between the predicted poses and the ground truth. As the MPJPE ignores the correlation across different joints, we introduce a novel loss function called joint crossover loss to consider the correlation across different joints. The final loss formulation is

$$\ell = \ell_m + \alpha\ell_j, \tag{5}$$

where ℓ_m denotes the MPJPE, α is a hyper-parameter which is set as 0.2 in our experiments, and ℓ_j denotes the joint crossover loss. For one training sample, the MPJPE is computed as [14]

$$\ell_m = \frac{1}{J(N+T)}\sum_{n=1}^{N+T}\sum_{j=1}^{J}\|\hat{p}_{j,n} - p_{j,n}\| \tag{6}$$

where $p_{j,n} \in \mathbb{R}^3$ is the predicted j-th joint position of the n-th human pose, $\hat{p}_{j,n}$ is the corresponding ground truth, and J is the number of joints.

We use joint crossover matrix Mat to compute the joint crossover loss. Specifically, for the predicted poses $P \in \mathbb{R}^{T\times K}$, we apply one convolutional layer with the kernel size $T\times 3$ to reduce the dimensionality of P, and obtain $P^{'}$, where $P^{'} \in \mathbb{R}^{J}$. $P^{'}$ indicates the joint information. The joint crossover matrix Mat is computed as:

$$Mat = P^{'T} \times P^{'}. \tag{7}$$

Here, each element in the matrix Mat represents the association information between two different joints. Finally, the joint crossover loss is computed as:

$$\ell_j = \frac{1}{J\times J}\sum_{i=1}^{J}\sum_{j=1}^{J}\left\|\hat{Mat}_{i,j} - Mat_{i,j}\right\|, \tag{8}$$

where Mat denotes the joint crossover matrix for the predicted poses, and $\hat{Mat}$ denotes the joint crossover matrix for the corresponding ground-truth.

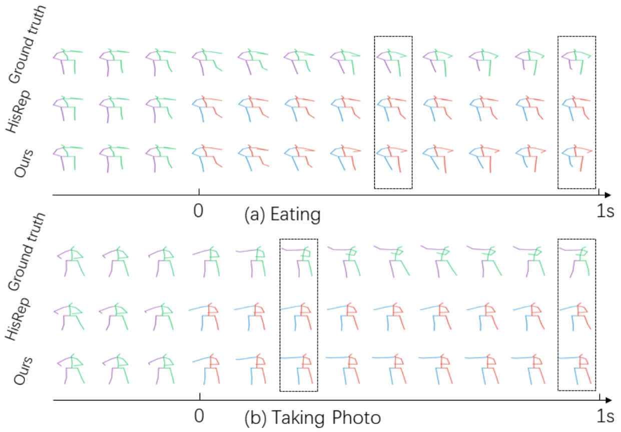

Fig. 2. Qualitative comparison of short-term and long-term predictions (”Eating” and ”Taking Photo”) on Human 3.6M dataset.

2.5. Implementation Details

We use the residual GCN model proposed in [13]. It consists of 12 residual blocks, each of which incorporates 2 graph convolutional layers with the hidden size 256. Besides, the GCN model contains two additional layers, one at the beginning and one at the end, to map the features to DCT coefficients to and decode the DCT coefficients to features, respectively. The pretrained f_q and f_k are both two-layer convolutional networks provided by [13] which can map the key vector to a 256 dimensional feature vector.

The learnable matrix $W^{(p)}$ for each layer p is of size 256×256 and the dimension of the adjacency matrix A is 66×66 for Human3.6M dataset and 75×75 for CMU Mocap dataset. We train using 50 clusters for each motion, observing 10 poses in the past and predicting up to 10 poses in the future. During test time, we predict 25 poses. Our network is trained for 40 epochs using a batch size of 32. We use an Adam optimizer with learning rate 0.0005 and 0.0025 weight decay.

3. EXPERIMENT

We evaluate the effectiveness of our model on two popular human motion datasets: Human 3.6M [15] and CMU Mocap [16]. To be consistent with previous work, we report our results for short-term ($<$ 500ms) and long-term($>$ 500ms) predictions. The experimental results of the compared methods are obtained by implementing their released code with their original settings.

Human 3.6M. Table 1 and Table 2 show the comparison results for short-term and long-term motion prediction on Human 3.6M, respectively. Note that our model outperforms all the compared methods on average prediction errors, which demonstrates the effectiveness of our method. Comparing with the Res.Sup [10] and convSeq2Seq [12], our method

Table 1. Comparison results of short-term motion prediction on Human 3.6M.

	Walking				Eating				Smoking				Discussion			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res.Sup [10]	23.2	40.9	61.0	66.1	16.8	31.5	53.5	61.7	18.9	34.7	57.5	65.4	25.7	47.8	80.0	91.3
convSeq2Seq [12]	17.7	33.5	56.3	63.6	11.0	22.4	40.7	48.4	11.6	22.8	41.3	48.9	17.1	34.5	64.8	77.6
HisRep [13]	**10.0**	**19.5**	**34.2**	**39.8**	**6.4**	**14.0**	28.7	36.2	**7.0**	14.9	29.9	36.4	10.2	23.4	52.1	65.4
Ours	10.3	19.7	34.6	40.1	6.6	14.2	**28.3**	**35.4**	7.2	**14.8**	**28.8**	**35.4**	**10.2**	**23.2**	**51.7**	**65.1**

	Directions				Greeting				Phoning				Posing				Purchases				Sitting			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res.Sup [10]	21.6	41.3	72.1	84.1	31.2	58.4	96.3	108.8	21.1	38.9	66.0	76.4	29.3	56.1	98.3	114.3	28.7	52.4	86.9	100.7	23.8	44.7	78.0	91.2
convSeq2Seq [12]	13.5	29.0	57.6	69.7	22.0	45.0	82.0	96.0	13.5	26.6	49.9	59.9	16.9	36.7	75.7	92.9	20.3	41.8	76.5	89.9	13.5	27.0	52.0	63.1
HisRep [13]	**7.4**	**18.4**	**44.5**	**56.5**	13.7	30.1	63.8	78.1	**8.6**	**18.3**	**39.0**	49.2	10.2	24.2	58.5	75.8	13.0	**29.2**	**60.4**	**73.9**	9.3	20.1	44.3	56.0
Ours	7.6	18.7	44.7	56.8	**12.9**	**28.9**	**62.0**	**76.5**	8.9	18.6	39.1	**49.0**	**9.7**	**23.4**	**57.3**	**73.9**	**12.7**	29.5	63.9	78.0	**9.3**	**19.9**	**43.1**	**54.6**

	Sitting Down				Taking Photo				Waiting				Walking Dog				Walking Together				Average			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res.Sup [10]	31.7	58.3	96.7	112.0	21.9	41.4	74.0	87.6	23.8	44.2	75.8	87.7	36.4	64.8	99.1	110.6	20.4	37.1	59.4	67.3	25.0	46.2	77.0	88.3
convSeq2Seq [12]	20.7	40.6	70.4	82.7	12.7	26.0	52.1	63.6	14.6	29.7	58.1	69.7	27.7	53.6	90.7	103.3	15.3	30.4	53.1	61.2	16.6	33.3	61.4	72.7
HisRep [13]	14.9	30.7	**59.1**	**72.0**	**8.3**	18.4	**40.7**	**51.5**	8.7	19.2	43.4	54.9	20.1	40.3	73.3	86.3	**8.9**	18.4	35.1	41.9	10.4	22.6	47.1	58.3
Ours	**14.8**	**30.6**	59.4	72.1	8.4	**18.4**	40.8	51.8	**8.6**	**18.5**	**41.2**	**52.1**	**18.4**	**38.9**	**71.9**	**84.3**	9.0	**18.3**	**34.2**	**41.0**	**10.3**	**22.4**	**46.7**	**57.7**

Table 2. Comparison results of long-term motion prediction on Human 3.6M.

	Walking				Eating				Smoking				Discussion			
milliseconds	560	720	880	1000	560	720	880	1000	560	720	880	1000	560	720	880	1000
Res.Sup [10]	71.6	72.5	76.0	79.1	74.9	85.9	93.8	98.0	78.1	88.6	96.6	102.1	109.5	122.0	128.6	131.8
convSeq2Seq [12]	72.2	77.2	80.9	82.3	61.3	72.8	81.8	87.1	60.0	69.4	77.2	81.7	98.1	112.9	123.0	129.3
HisRep [13]	47.4	52.1	55.5	58.1	50.0	61.4	70.6	75.7	47.6	56.6	64.4	69.5	86.6	102.2	113.2	119.8
Ours	**47.2**	**52.0**	**55.9**	**58.7**	**48.2**	**59.1**	**68.2**	**73.3**	**67.0**	**82.4**	**96.2**	**68.9**	**85.0**	**99.8**	**108.5**	**118.3**

	Directions				Greeting				Phoning				Posing				Purchases				Sitting			
milliseconds	560	720	880	1000	560	720	880	1000	560	720	880	1000	560	720	880	1000	560	720	880	1000	560	720	880	1000
Res.Sup [10]	101.1	114.5	124.5	129.1	126.1	138.8	150.3	153.9	94.0	107.7	119.1	126.4	140.3	159.8	173.2	183.2	122.1	137.2	148.0	154.0	113.7	130.5	144.4	152.6
convSeq2Seq [12]	86.6	99.8	109.9	115.8	116.9	130.7	142.7	147.3	77.1	92.1	105.5	114.0	122.5	148.8	171.8	187.4	111.3	129.1	143.1	151.5	82.4	98.8	112.4	120.7
HisRep [13]	73.9	88.2	100.1	106.5	101.9	118.4	132.7	138.8	67.4	82.9	96.5	**105.0**	107.6	136.8	161.4	178.2	**95.6**	**110.9**	**125.0**	**134.2**	76.4	93.1	107.0	115.9
Ours	**73.7**	**87.5**	**98.4**	**104.5**	**99.8**	**115.0**	**127.3**	**132.2**	**67.0**	**82.4**	**96.2**	105.2	**103.7**	**130.0**	**150.4**	**164.7**	99.6	115.9	130.3	139.8	**75.2**	**91.5**	**105.1**	**113.7**

	Sitting Down				Taking Photo				Waiting				Walking Dog				Walking Together				Average			
milliseconds	560	720	880	1000	560	720	880	1000	560	720	880	1000	560	720	880	1000	560	720	880	1000	560	720	880	1000
Res.Sup [10]	138.8	159.0	176.1	187.4	110.6	128.9	143.7	153.9	105.4	117.3	128.1	135.4	128.7	141.1	155.3	164.5	80.2	87.3	92.8	98.2	106.3	119.4	130.0	136.6
convSeq2Seq [12]	106.5	125.1	139.8	150.3	84.4	102.4	117.7	128.1	87.3	100.3	110.7	117.7	122.4	133.8	151.1	162.4	72.0	77.7	82.9	87.4	90.7	104.7	116.7	124.2
HisRep [13]	**97.0**	**116.1**	**132.1**	**143.6**	**72.1**	**90.4**	105.5	**115.9**	74.5	89.0	100.3	108.2	108.2	120.6	135.9	146.9	52.7	57.8	62.0	64.9	77.3	91.8	104.1	112.1
Ours	97.5	117.3	133.6	144.9	72.5	90.7	**105.5**	116	**71.1**	**84.8**	**95.4**	**102.9**	**103.4**	**114.4**	**128.9**	**139**	**50.5**	**55.3**	**59.3**	**61.8**	**76.2**	**90.3**	**102.0**	**109.6**

Table 3. The average results for short-term and long-term motion prediction on CMU Mocap.

milliseconds	80	160	320	400	1000
LearnTraj [14]	11.2	21.3	38.2	45.5	**80.9**
HisRep [13]	8.8	17.8	**30.1**	**37.4**	85.1
Ours	**8.5**	**16.0**	32.4	39.2	83.8

Table 4. Ablation study. It shows the average prediction results on Human 3.6M.

keysequences	crossover loss	80	160	320	400	1000
✓		10.4	22.5	46.9	57.9	109.9
	✓	10.3	22.5	47.1	58.1	110.0
✓	✓	**10.3**	**22.4**	**46.7**	**57.7**	**109.6**

achieves great improvements across all the motions. Besides, our method achieves higher accuracy on some motions with clear repeated patterns, such as ”Walking Dog” and ”Walking Together”. In the same time, our method remains competitive on the other motions. We also provide qualitative comparisons in Fig. 2. We can see that poses predicted by our method are closed to the ground truth.

CMU Mocap. Table 3 shows the average results of short-term and long-term prediction on CMU Mocap. Note that we perform our experiment on 4 actions: running, soccer, walking and washwindow. Our method consistently outperforms the compared methods, which demonstrates the advantages of our key subsequences and the crossover loss.

3.1. Ablation Study

As shown in Table 4, we respectively remove the key subsequences and the crossover loss to verify their effectiveness. The results shows that the key subsequences and the crossover loss both contribute to the improvement of the performance.

4. CONCLUSION

In this paper, we propose to predict human motion based on key subsequences extracted from the whole motion history by clustering, which can help to fully employ the information contained in the whole motion history. We also introduce joint crossover loss, which can learn the dependencies across different joints, to train the model. Experimental results on public human motion datasets show that our method achieves better performance over state-of-the-art methods in motion prediction, which demonstrates the effectiveness of our method.

5. ACKNOWLEDGMENTS

This work was supported by the Natural Science Foundation of China (NSFC) under Grant No.61972038.

6. REFERENCES

[1] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[2] Hema Swetha Koppula and Ashutosh Saxena, "Anticipating human activities for reactive robotic response.," in *IROS*. Tokyo, 2013, p. 2071.

[3] Lucas Kovar, Michael Gleicher, and Frédéric Pighin, "Motion graphs," in *ACM SIGGRAPH 2008 classes*, pp. 1–10. 2008.

[4] Sergey Levine, Jack M Wang, Alexis Haraux, Zoran Popović, and Vladlen Koltun, "Continuous character control with low-dimensional embeddings," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–10, 2012.

[5] Hedvig Sidenbladh, Michael J Black, and Leonid Sigal, "Implicit probabilistic models of human motion for synthesis and tracking," in *European conference on computer vision*. Springer, 2002, pp. 784–800.

[6] Matthew Brand and Aaron Hertzmann, "Style machines," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 183–192.

[7] Jack M Wang, David J Fleet, and Aaron Hertzmann, "Gaussian process dynamical models for human motion," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 283–298, 2007.

[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*, 2015, pp. 4346–4354.

[9] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2016, pp. 5308–5317.

[10] 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*, 2017, pp. 2891–2900.

[11] Judith Butepage, Michael J Black, Danica Kragic, and Hedvig Kjellstrom, "Deep representation learning for human motion prediction and classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6158–6166.

[12] 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*, 2018, pp. 5226–5234.

[13] Wei Mao, Miaomiao Liu, and Mathieu Salzmann, "History repeats itself: Human motion prediction via motion attention," in *European Conference on Computer Vision*. Springer, 2020, pp. 474–489.

[14] 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*, 2019, pp. 9489–9497.

[15] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu, "Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2013.

[16] Timo von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 601–617.