3D Human Motion Prediction with Recurrent Networks and Motion Discriminator

Annamalai Lakshamanan
alakshmanan@student.ethz.ch

Arda Arslan
aarslan@student.ethz.ch

Gökberk Özsoy
goezsoy@student.ethz.ch

Gowtham Senthil
gsenthi@student.ethz.ch

ABSTRACT

Human motion prediction is the complex task of predicting future 3D locations of human skeletal joints given a past motion. In this work, we model the spatial dimension (interaction of joints) with a graph, and the temporal dimension (evaluation of motion pattern) with a recurrent network. Moreover, we employ adversarial training to ensure that our forecast is consistent and realistic. We evaluate our solution against multiple state-of-the-art baselines, and claim the new best for some categories on Human 3.6M and AMASS datasets.

1 INTRODUCTION

3D human motion prediction is the task of predicting future human-joint locations given an initial motion sequence. It has a wide-range of applications in human-robot interaction [20], autonomous driving [16], and robotics [9] in addition to generative applications in computer graphics, animation and game development [10]. While it has been extensively studied for short-term prediction of simple human activities such as walking, smoking, and sitting, realistic long-term prediction is still a daunting problem.

Main challenges behind the task are the context-dependent, highly dynamic, and non-linear stochastic nature of human motion, and the complex relation between joints. However, with the availability of large datasets [11] and the advent of high quality research in deep learning such as GANs [6] and GCNs [18], there has been considerable development in human motion prediction.

In this paper, we propose an end-to-end trainable pipeline, where we first encode the given motion sequence using graph convolutional networks, and then decode the future poses using recurrent networks with and without attention support. Moreover, we utilize an adversarial training, where the discriminator tries to distinguish real and predicted sequences.

In human skeleton, each joint is dependent on the connecting joints locally, as well as unconnected ones during the motion pattern globally. Thus, they can be modelled as nodes in a kinematic tree [21], and a graph neural network is a natural choice to embed information using this tree. Here, we use STS-GCN [18] model as our encoder. In the decoding process, we initialize our recurrent network with the encoding, and auto-regressively predict the output sequence. For some models, we add attention to select the most relevant frame among the input for the current prediction. Furthermore, discriminator acts as a motion prior and regularizes the encoder-decoder by assessing the reality of the forecast.

We trained and tested our model and its variants on Human 3.6M [7] and AMASS [11] datasets. As apparent from the evaluation, depending on the dataset and forecasting length, we get scores that are slightly above the state-of-the-art, proving our model to be promising.

2 RELATED WORK

As this is a sequential task by nature, recurrent networks were employed in early works. Fragkiadaki et al. [5] used LSTM based encoder-decoder to predict the next instances. However, this was found to be effective only for short term prediction and it later collapses to static frames. Martinez et al. [15] proposed velocity forecasting by adding residual connections from previous prediction to current. In addition, they introduced scheduled sampling where they forfeit teacher forcing during training.

Aksan et al. [2] discuss the lack of explicit spatial modelling in existing works, and extends [15] by including a structured prediction layer after recurrent layer to decompose pose prediction into individual joint prediction. To cope with static pose convergence, Aksan et al. [1] proposed a model that learns high dimensional embeddings for joints followed by a decoupled temporal and spatial self-attention mechanism. Zhang et al. [22] proposed an architecture with a GRU encoder that maps motion information into different frequency bands using DCT. The decoder, performs inverse DCT and uses GRUs to predict in temporal domain.

Mao et al. [14] utilize Graph Convolutional Networks (GCN) to learn inter-joint dependencies. Further, Mao et al. [12] propose the concept of motion attention to capture the similarity between current motion context and historical motion sub-sequences with a post-processing using GCN. Sofianos et al. [18] proposed a STS-GCN, to address both the underlying spatial and temporal correlations. Here, a spatio-temporal-separable graph convolutional encoder followed by a temporal convolutional decoder is used to predict the human poses. Despite being a lightweight model architecture, it is able to predict long-term motion (1000 ms) from a small input seed sequence (400 ms).

Bem et al.[4] propose to disentangle the poses and appearance so that poses and appearances can be manipulated independently by GAN. It consists of four layered architecture with encoder, prior, decoder and finally a discriminator to predict future instances. In [8], Kocabas et al. use a motion prior model based on VAE to learn plausible motions. A GRU based temporal generator trained together with an attention based motion discriminator is used to generate new poses and their SMPL meshes for video generation.

3 METHOD

Let us now explain our solution for the problem. A motion sequence $\mathbf{X}_{1:T} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ consists of a series of frames $\mathbf{x}_t \in \mathcal{R}^N$, $1 \leq t \leq T$, each containing 3D angle or position information of N joints.

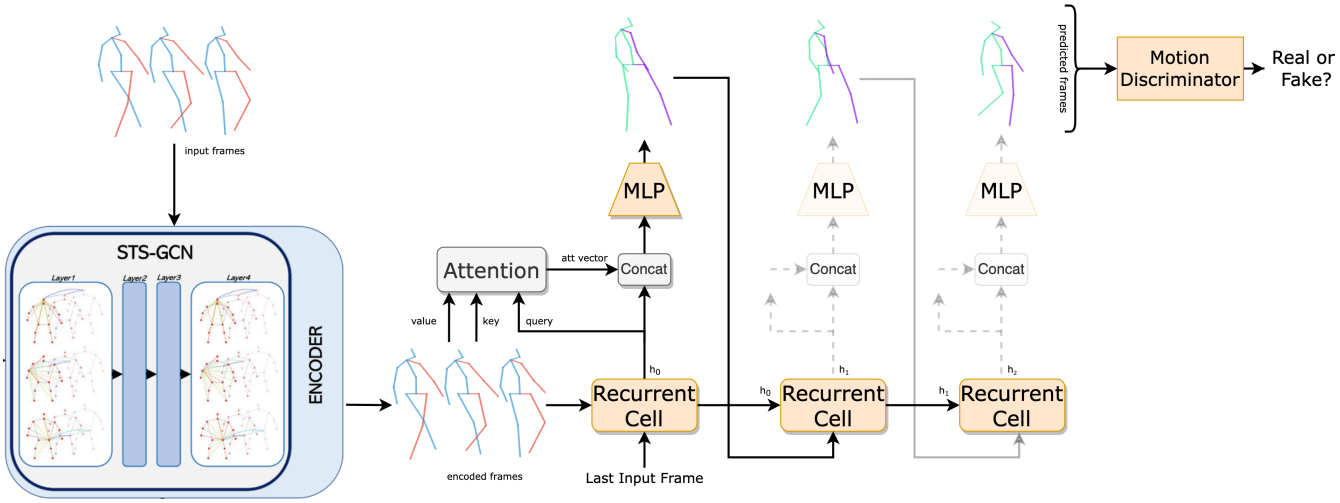

Figure 1: Proposed model architecture. For RNN-only model, only orange colored parts of the decoder are present. For RNN with Attention model, both orange and gray parts are included. Zoom in for labels.

Our aim is to develop a predictive model that maps $\mathbf{X}_{1:T}$ to future sequences $\tilde{\mathbf{X}}_{T+1:T+\Delta} = \{\tilde{\mathbf{x}}_{T+1}, \tilde{\mathbf{x}}_{T+2}, \ldots, \tilde{\mathbf{x}}_{T+\Delta}\}$ that is similar to the ground truth $\mathbf{X}_{T+1:T+\Delta}$, where Δ is the length of prediction. At this point, our model consists of three main parts: (i) encoding features from given seed sequence $\mathbf{X}_{1:T}$ via STS-GCN [18], (ii) decoding future frames $\tilde{\mathbf{X}}_{T+1:T+\Delta}$ using encoded information via attention based recurrent networks, and (iii) providing a regularization with a motion discriminator. A detailed diagram for the whole pipeline can be seen in Figure 1.

3.1 Encoder: Space-Time Separable GCN [18]

We define graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where TN nodes are all body joints across $\mathbf{X}_{1:T}$, and edges are the elements of adjacency matrix $A^{st} \in \mathcal{R}^{TNxTN}$ representing interaction of all joints at all times. Now, the spatio-temporal relations of joints for given seed sequence can be encoded by a graph convolutional network $f(\mathbf{X}_{1:T}, A^{st}, W)$, where W represents the network weights, and A represents the learnable adjacency matrix. Each layer l of f outputs $H^{l+1} \in \mathcal{R}^{C^{l+1}xNxT}$ with the following computation:

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

where σ is PReLU, $W^l \in \mathcal{R}^{C^l x C^{l+1}}$, $H^1 = \mathbf{X}_{1:T}$, and $C^1 = 3$ as each joint is in 3D.

In [18], authors propose to factorize A^{st} matrix into A^t and A^s to reduce number of trainable parameters and increase forecasting performance. This way, they only allow talk of joints in the same frame (A^s), and talk of same joint across different frames (A^t), preventing the unnecessary cross talk. Now, each layer l of f is computed by the following:

$$H^{l+1} = \sigma(A^{s\,l} A^{t\,l} H^l W^l) \tag{2}$$

Eq.2 represents only a single layer. In total, there are four such layers with residual connections. Each added layer increases the number of joints that are contributed to the final embedding of a particular joint.

3.2 Decoder: RNN with Attention

Having a robust encoder, we generated our prediction in an auto-regressive fashion using recurrent cells, LSTM and GRU. We tried out two types of models.

RNN only. All encoded information (output of STS-GCN encoder) is flattened and given to recurrent cell as the initial context vector. The first input to this recurrent cell is the last frame of seed sequence. Then, the hidden state of the recurrent cell is fed to an MLP to receive the first frame prediction. Following [15], we used scheduled sampling as opposed to teacher forcing, meaning that we give the predicted frame as the next input to the recurrent cell even during training. Until the requested number of future frames, the same procedure is applied.

RNN with Attention Mechanism[19]. Attention enables the decoder to query encoded information from all time-steps as opposed to RNN-only case. Thus, this time, only the last encoded frame information is given to recurrent cell as initial context vector and its first input is the last frame of seed sequence. Then, the hidden state of the recurrent cell is used as a query, while each frame encoding is used as both key and value in the attention mechanism. As most of the motion patterns are repetitive in time, we hypothesize that the model will attend to the most relevant encoded frame from input for its prediction. The computed attention vector is then concatenated with the hidden state of the recurrent cell and fed to an MLP to give final prediction. Similar to the RNN only model, we use scheduled sampling to obtain the requested number of future frames.

3.3 Motion Discriminator

To regularize predicted motion sequences, we used a motion discriminator as in [8]. Our discriminator has a similar architecture with the decoder part of [18]. It consists of two convolutional layers followed by a linear layer and a sigmoid function. The discriminator makes a prediction between 0 and 1, where 0 indicates the image is generated by the generator network and 1 indicates the image

is real. We used binary cross entropy loss for both training the discriminator and for the loss signal which is provided to generator by the discriminator. To avoid the discriminator to be too confident in its predictions, we used two-sided label smoothing, i.e. during training we replaced the labels which correspond to true images with 0.8 and the labels which correspond to fake images with 0.2. Loss functions used during adversarial training is as following:

$$L_{adv} = -\mathbb{E}_{x\sim p_g(x)}[0.8 * \log D(x) + 0.2 * \log(1 - D(x))] \quad (3)$$

$$L_{disc} = -\mathbb{E}_{x\sim p_d(x)}[0.8 * \log D(x) + 0.2 * \log(1 - D(x))] - \mathrm{E}_{x\sim p_g(x)}[0.2 * \log D(x) + 0.8 * \log(1 - D(x))] \quad (4)$$

3.4 Implementation Details

The proposed architecture is trained end-to-end using Mean Per Joint Position Error (MPJPE) [7], which is defined as follows:

$$L_{MPJPE} = \frac{1}{N\Delta} \sum_{t=T+1}^{T+\Delta} \sum_{j=1}^{N} ||\tilde{\mathbf{x}}_{t,j} - \mathbf{x}_{t,j}||^2 \quad (5)$$

where $\tilde{\mathbf{x}}$ is the prediction and $\mathbf{x}$ is the ground truth.

Our encoder has 4 layers with different channel sizes C^l, starting from 3 to 64, then 64 to 32, then 32 to 64, and finally 64 to 3. This means that the seed sequence and STS-GCN output have the same shapes. We used 3 heads in our multi-head attention mechanism, with query, key, and value shapes all equal to $3xN$. At each layer of encoder, we used batch normalization and residual connections. In addition, for RNN with Attention model, we used residual connection [15], where we added the output of MLP and the last predicted frame (also current step's recurrent cell input). For more model details, kindly refer the configuration file in attached codebase.

In all our experiments, we used the same data preprocessing as in [18] and [13] to be comparable with the literature. We used Adam as optimizer, different learning rate, batch size and learning rate decay regimes for different experiments. More details can be found in our codebase. All our experiments were implemented using PyTorch.

To train the discriminator network, we used Adam optimizer, learning rate 0.01, gradient clipping at norm 5.0, multi-step learning rate scheduling with gamma 0.1 and milestones [15, 25, 35, 40]. To stabilize the overall training, we multiplied the loss signal provided to generator by the discriminator by 0.01.

4 EVALUATION

4.1 Datasets

Human 3.6M [7]. This dataset consists of 3.6 million 3D poses and images of 11 humans performing 17 different actions. Out of the 32 joints, we considered 22 joints for our models similar to the joints used in [18]. We ignore the other 10 joints as they are constant or at same position with respect to other joints. Out of the 17 actions in total, we use only 15 different actions. Among subjects, we use subject 5 for testing, subject 11 for validation and all the other five subjects for training. In order to get the canonicalized coordinates, we first remove global rotation and orientation. Then using euler-rodrigues formula we get the rotation matrix, thereby the 3D pose. This is done similar to the procedure in [18] and [13].

AMASS [11]. The Archive of Motion Capture as Surface Shapes (AMASS) contains a total of 18 motion capture datasets. Among them, we consider 13 datasets with 8 for training, 4 for validation and 1 for testing (BMLrub). Like H3.6M, we consider the selection as done in [18] and [13]. The motion capture data consists of 52 joints including body and hand joints. For the motion we attempt to predict, we require only the 22 body joints. As 4 joints lead to static pose, we finally require only 18 joints and estimate their pose only. Here also we do canonicalization the same way we proceed for H3.6M dataset.

In our experiments, we used a seed motion sequence of length 400ms (10 frames) to predict the subsequent 400ms (10 frames) and 1000ms (25 frames) for both AMASS and H3.6M datasets.

4.2 Evaluation Metrics and Baselines

For evaluation, we report the Mean Per Joint Position Error (MPJPE) in milimeter, which is commonly used in human motion prediction.

Baseline models are as follows: Zero-Velocity model is proposed in [15], where the output is created by simply repeating the last frame of the input sequence. It is a powerful baseline as it surpassed most of the models when it was published. Simple LSTM and GRU models are using recurrent networks for both encoder and decoder. STSGCN + TCN model is the default model proposed in [18], where they use temporal convolutional networks for keeping the parameter size small.

4.3 Results

Now, we will present test results about our experiments for Human 3.6M (Table 1), and AMASS (Table 2) datasets. We report the average MPJPE scores of all predictions regardless of the action and sub-dataset type. For the models which include '+' in its name, left side describes encoder, and the right side describes decoder. If we used discriminator as well, it is indicated by '+ Motion Disc'. The models which have '*' sign are the baselines we mentioned in section 4.2. Other models are the variants of our own model. As can be seen, for both datasets, and both short(400 msec) and long(1000 msec)-term forecasts, our proposed models' variants hold the lowest MPJPE scores.

Model Architecture	MPJPE (mm)	
	400 ms	1000 ms
Zero Velocity *	54.60	93.33
Simple LSTM *	47.52	84.58
Simple GRU *	48.50	83.95
STSGCN + TCN *	42.26	77.27
STSGCN + TCN + Motion Disc	42.45	**77.04**
STSGCN + LSTM	**35.34**	79.47
STSGCN + GRU	37.13	81.42
STSGCN + Attention GRU	37.93	82.13

Table 1: Human 3.6M results for short (400 msec) and long (1000 msec) term motion prediction

Model Architecture	MPJPE (mm)	
	400 ms	1000 ms
Zero Velocity *	41.68	87.78
Simple LSTM *	27.08	47.72
Simple GRU *	26.97	47.31
STSGCN + TCN *	24.30	44.32
STSGCN + TCN + Motion Disc	27.54	43.77
STSGCN + LSTM	**20.71**	42.12
STSGCN + GRU	24.50	**41.68**
STSGCN + Attention GRU + Motion Disc	32.76	57.99

Table 2: AMASS results for short (400 msec) and long (1000 msec) term motion prediction

5 DISCUSSION

After doing a literature review, we observed that the state-of-the-art performance for human motion estimation task was achieved by [18]. We refactored their repository into a more structured version and reproduced the results on the paper. Then by keeping the data pre-processing and evaluation methods fixed, we tried different architectures from other works. As an initial attempt, we tried the VAE-DCT architecture from [22]. However, this architecture yielded high MPJPE values and we concluded that such a generative architecture is more suitable for novel human motion synthesis tasks rather than human motion estimation tasks. We also tried the Structured Prediction Layer which was proposed in [2], with different RNN architectures, however we could not manage to get a significant improvement. We were convinced that STS-GCN was a powerful architecture for this task and we decided to build our experiments on top of it.

During adversarial training, we initially tried an STS-GCN encoder as the discriminator architecture, however this was a very powerful architecture, and it was too certain with its predictions already at the beginning of training, and this caused the gradients provided to the generator to be too small. To overcome this issue, we added Gaussian noise to inputs of the discriminator, but this destabilized the training. Therefore, we decided to use a small convolutional neural network as the discriminator architecture and multiply the loss provided to the generator by the discriminator by a small value (0.01). This was the only case when we got a stabilized training.

Since the beginning of our adversarial experiments, we smoothed the labels in a two-sided way as explained in Section 3.3. After conducting our experiments, we learned that, it is suggested to use one-sided label smoothing instead, for example in [17]. However, we did not have enough time to rerun our experiments.

We also realized that in most of the experiments, discriminator was still powerful compared to the generator since it was very successful at distinguishing real and generated samples. And this might cause the gradients provided by discriminator to generator to be too small. We could have tried to use a Wasserstein GAN to overcome this issue, however again we did not have time for it.

During training STSGCN + Attention GRU, which is an autoregressive architecture, we observed that using teacher forcing instead of feeding previous predictions as input to next time steps helped with the convergence of training loss. However, this yielded high validation loss values. To overcome this issue, we used scheduled sampling for this experiment as suggested in [3]. In the same experiment, we noticed that if we do not use a residual connection between input and output of a GRU cell (as in [15]), then the model predictions were always the same across different timesteps.

Although we expected the attention mechanism to improve the results, it turned out that for this task, attentively querying information from encoded motion sequence or the previous predictions did not give a better result than using a fixed sized hidden state which represents all the previous information.

6 CONCLUSION

Human motion prediction is about discovering spatio-temporal relations between joints. Thus, a GCN powered encoder, and a recurrence based decoder are natural candidates. Combined with various tricks on the decoder side, and adversarial training, we present new state-of-the-art performances for short and long term horizons.

REFERENCES

[1] Emre Aksan, Peng Cao, Manuel Kaufmann, and Otmar Hilliges. 2020. Attention, please: A Spatio-temporal Transformer for 3D Human Motion Prediction. *CoRR* abs/2004.08692 (2020). arXiv:2004.08692 https://arxiv.org/abs/2004.08692

[2] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. 2019. Structured Prediction Helps 3D Human Motion Modelling. *CoRR* abs/1910.09070 (2019). arXiv:1910.09070 http://arxiv.org/abs/1910.09070

[3] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. arXiv:cs.LG/1506.03099

[4] Rodrigo Andrade de Bem, Arnab Ghosh, Thalaiyasingam Ajanthan, Ondrej Miksik, N. Siddharth, and Philip H. S. Torr. 2018. DGPose: Disentangled Semi-supervised Deep Generative Models for Human Body Analysis. *CoRR* abs/1804.06364 (2018). arXiv:1804.06364 http://arxiv.org/abs/1804.06364

[5] Katerina Fragkiadaki, Sergey Levine, and Jitendra Malik. 2015. Recurrent Network Models for Kinematic Tracking. *CoRR* abs/1508.00271 (2015). arXiv:1508.00271 http://arxiv.org/abs/1508.00271

[6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. arXiv:stat.ML/1406.2661

[7] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2014. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 7 (jul 2014), 1325–1339.

[8] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. 2019. VIBE: Video Inference for Human Body Pose and Shape Estimation. *CoRR* abs/1912.05656 (2019). arXiv:1912.05656 http://arxiv.org/abs/1912.05656

[9] Hema Swetha Koppula and Ashutosh Saxena. 2013. Anticipating human activities for reactive robotic response. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2071–2071. https://doi.org/10.1109/IROS.2013.6696634

[10] Sergey Levine, Jack M. Wang, Alexis Haraux, Zoran Popović, and Vladlen Koltun. 2012. Continuous Character Control with Low-Dimensional Embeddings. *ACM Trans. Graph.* 31, 4, Article 28 (jul 2012), 10 pages. https://doi.org/10.1145/2185520.2185524

[11] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. 2019. AMASS: Archive of Motion Capture as Surface Shapes. In *International Conference on Computer Vision*. 5442–5451.

[12] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. 2020. History Repeats Itself: Human Motion Prediction via Motion Attention. *CoRR* abs/2007.11755 (2020). arXiv:2007.11755 https://arxiv.org/abs/2007.11755

[13] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. 2020. History Repeats Itself: Human Motion Prediction via Motion Attention. *CoRR* abs/2007.11755 (2020). arXiv:2007.11755 https://arxiv.org/abs/2007.11755

[14] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. 2019. Learning Trajectory Dependencies for Human Motion Prediction. *CoRR* abs/1908.05436 (2019). arXiv:1908.05436 http://arxiv.org/abs/1908.05436

[15] Julieta Martinez, Michael J. Black, and Javier Romero. 2017. On human motion prediction using recurrent neural networks. *CoRR* abs/1705.02445 (2017). arXiv:1705.02445 http://arxiv.org/abs/1705.02445

[16] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. 2016. A survey of motion planning and control techniques for self-driving urban

vehicles. *IEEE Transactions on intelligent vehicles* 1, 1 (2016), 33–55.

[17] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved Techniques for Training GANs. *CoRR* abs/1606.03498 (2016). arXiv:1606.03498 http://arxiv.org/abs/1606.03498

[18] Theodoros Sofianos, Alessio Sampieri, Luca Franco, and Fabio Galasso. 2021. Space-Time-Separable Graph Convolutional Network for Pose Forecasting. *CoRR* abs/2110.04573 (2021). arXiv:2110.04573 https://arxiv.org/abs/2110.04573

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[20] Erwin Wu and Hideki Koike. 2020. FuturePong: Real-Time Table Tennis Trajectory Forecasting Using Pose Prediction Network. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3334480.3382853

[21] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (Apr. 2018). https://ojs.aaai.org/index.php/AAAI/article/view/12328

[22] Yan Zhang, Michael J. Black, and Siyu Tang. 2020. We are More than Our Joints: Predicting how 3D Bodies Move. *CoRR* abs/2012.00619 (2020). arXiv:2012.00619 https://arxiv.org/abs/2012.00619