The Best TIM: Human Motion Prediction

Fernando Gonzalez
fgonzalez@student.ethz.ch

Cristina Guzmán
csolis@student.ethz.ch

Zixin Shu
zixshu@student.ethz.ch

ABSTRACT

In this work, we tackle the problem of predicting human motion, where we are asked to predict 400 milliseconds of motion given 2 seconds of pre-recorded data. We build upon an approach in which body joint trajectories are encoded at different temporal scales. We explore some variants and implementations of this approach and show that making use of shorter sequences and augmenting the dataset instead can be beneficial, improving the performance in terms of mean joint angle difference loss. Furthermore, we discuss additional architectures that make use of adversarial learning that can potentially lead to also better results.

1 INTRODUCTION

Understanding the environment by which an individual is surrounded, and being able to predict scenarios in such environment is an ability of utmost importance in a world where human-machine interaction is increasingly becoming part of our daily lives. Human motion prediction is a stepping stone towards such understanding, and thus has become a classical problem in the field of computer vision. The goal is to predict future frames of human motion skeletons given past observed frames. It has applications in a huge variety of tasks such as in robotics and augmented reality.

Different approaches to tackle the problem of predicting human motion have been proposed, including work done under the regime of recurrent neural networks [1][3][7] , this given the sequential nature of the problem at hand.

Other approaches nevertheless, have considered other kinds of architectures, namely using graph convolutional networks (GCN) for capturing spatial dependencies of the human joints [6]. In addition to this, [4] propose feeding into the GCN an encoding of the sequences at different temporal scales by using what they call temporal inception modules.

There is also recent literature that proposes attention-based architectures, given the observation that human motion tends to repeat itself.[5]

We build upon the work of [4], and our main contributions are three fold:

(i) We adjust the overall architecture to align to our task, namely using longer sequences with length of 26 frames and adding more filters for each sequence length.
(ii) We extend the TIM architecture by adding more different-sized 1D convolutional filters and change the dilation and stride for the filters of the longer sub-sequences, in order to detect higher-level patterns.
(iii) We augment the given dataset by sampling two sequences per observation.

We show that the used models can benefit from the implementation we propose. Additionally, we discuss an interesting approach within the framework of adversarial learning, similar to what is done in [8].

2 METHOD

2.1 Task Definition

In this task we are given 120 frames (2 seconds) of human motion to predict the future 24 frames (400 milliseconds) represented with rotation matrices from 15 joints, i.e. 135 joint position features at every time step.

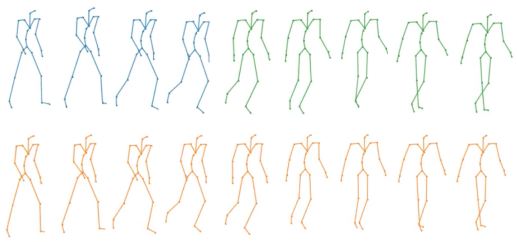

Figure 1: Example of visualization of predicted (blue and green) and ground truth frames (orange)

2.2 Model Overview

We are using a variant of the model proposed by [4] (TIM Model). There, they use a GCN that learns the residuals between the target sequence and the input sequence with the last pose replicated 24 times. As input for the GCN, they use embeddings produced by a Temporal Inception Module (TIM). TIM encodes the motion of the most recently seen frames by convolving such frames with 1D filters of different sizes, proportional to the sequence length.

We propose 3 main modifications to [4].

Data augmentation. We extracted two samples per sequence given and concatenated them into one big training set, leading to the size of the training set doubled. Also, we used a total number of $M_J = 26$ frames to predict their future $F = 24$ frames. As it is shown in Figure 3, the first set of M_J frames is extracted from the beginning of every sequence and the second one is selected at random with the beginning of sequence index being uniformly distributed between $M_J + F^*$ onward.

Using different sequence size. We increased the size of the observed length, which is the number of frames we look back at the TIM layer, from 5 and 10 increased to 13 and 26.

*In order to avoid overlap between test frames from the first sub-sequence and training frames from the second one

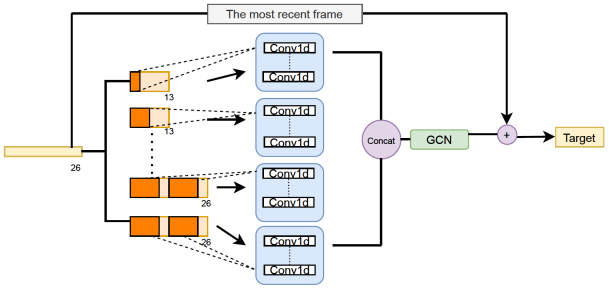

Figure 2: Model pipeline. Every joint position representation passes through our model pipeline to obtain its corresponding target sequence. Each block of 1D convolutional filters has filters of the same size, each block consisting of different filter size and number of filters, these being proportional to the subsequence lengths. The filters convolving with 26-long subsequences applied dilation and stride. Dark orange boxes in the 26-long sub-sequences represent the dilation and stride.

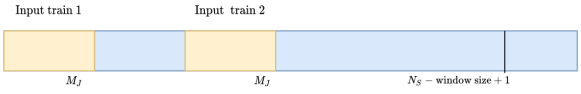

Figure 3: Data augmentation procedure. The first subsequence corresponds to the first M_J frames and the second one is a random sample of M_J consecutive frames. N_S is the length of the original raw sequence and window size = 144.

Architecture. Regarding TIM, we changed the number of convolutional filter sizes from 5 to 9 with dilation and stride in all filters that are convolved with the sub-sequences of length 26. The details of the architecture of the Temporal Inception Module we used in our submission are shown in Table 1. We also modified the GCN by increasing the number of layers in the graph convolutional block from 2 to 3.

Sub-sequence input length	Number of filter	filter size	Dilation, stride
13	12	3	-
13	9	4	-
13	9	5	-
13	9	6	-
26	9	5	2,2
26	7	6	2,2
26	6	8	2,2
26	5	10	2,2
26	4	12	2,2

Table 1: Details of TIM module in the submission

Implementation details. As additional details, we also found out that decreasing the learning rate while training helps the predictions improve further. Also, a increasing batch size helps to improve the score up to some point where it gets stuck or throws cuda memory errors. The batch size in our final submission is 32.

The loss used for training is Mean Per Joint Position Error (MPJPE) as proposed in [2] and is defined as

$$\frac{1}{N(M_J+F)} \sum_{f=-M_J}^{F-1} \sum_{i=1}^{I} \left\| \mathbf{p}_{i,f} - \hat{\boldsymbol{p}}_{i,f} \right\|^2 \tag{1}$$

where $\boldsymbol{p}_{i,f}$ and $\hat{\boldsymbol{p}}_{i,f}$ are the rotation matrices corresponding to join trajectories for the $i-th$ joint in $f-th$ frame in the ground truth and predicted sequences, respectively, M_J is the number of past frames, F is the number of future frames, and N is the number of joint trajectories in total, so in our case $M_J = 26$, $F = 24$, $I = 15$, $N = 135$.

3 EVALUATION

In this section, we present the experiments we tried, the motivation behind them, and the results that support the choice of such modifications. We assess the performance of the different experiments by means of the joint angle difference loss in the validation set.

We initially claimed that using shorter sequences was enough for predicting 24 frames, and based our guess of using 26 frames on the visualization of some of the sequences. Because of this, we decided that it would be more beneficial to reduce the sequence length and increase the sample size instead. We tested however longer sequence lengths (48) but the results were not better.

Regarding the modifications done to the architecture, we increased the number and sizes of the filters, given that we are dealing with larger sequences compared to [4] who use 5 different filter size configurations. We also implemented stride and dilation to all the filters used for convolving with the longer subsequences, in order capture higher-level patterns in the sequence while increasing the receptive field and thus looking at more spread frames.

The results ** of the experiments are provided in Table 4. They show that the data augmentation, the changes to the sequence length (10 to 26) and to the architecture provide an improvement in performance. We observe that the improvements across different timestamps in the validation set increase for further frames.

	Val 5	Val 10	Val 24	Test
A-seq2seq	0.3089	1.012	4.324	3.783
A-GCN	0.149	0.521	2.552	2.125
TIM	0.150	0.535	2.716	2.315
Ours	**0.1306**	**0.4676**	**2.405**	**1.874**

Table 2: Performance comparison joint angle difference

** The first two models shown correspond to the additional adversarial models that we tried, which will be discussed in the next section.

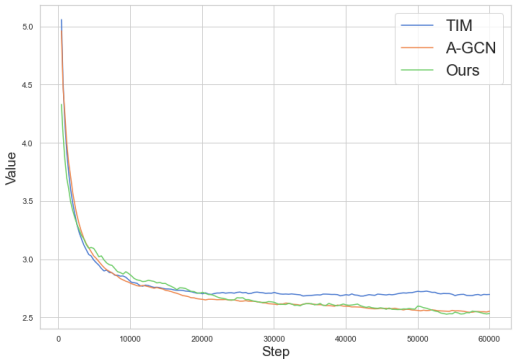

Figure 4: Mean joint angle difference until frame 24 for TIM Model, Adversarial GCN and Our model

4 DISCUSSION

4.1 Extended TIM

TIM was originally implemented in both Human3.6M and CMU motion capture dataset. In our experiment, the architecture achieved a good performance on AMASS dataset using rotation matrices. A possible extension would be transforming the data into 3D positions since recent approaches have pointed out that angle representations suffer from ambiguities and they addressed this issue with 3D joint position representations.

The number of sub-sequence samples that we are using for TIM as well as the number of stages and GCN layers can be optimized with a grid search. Here, after a series of experiments we provide the combination of the parameters that yield better results and an intuition of the change in performance of the model corresponding to the modification of these parameters.

As shown in the evaluation section, the performance of our model improves compared to the original TIM model, with differences between their scores increasing as we predict further frames. This can be due to the amount of additional information captured given the increase in the receptive fields when using dilated convolutions.

For the evaluation of experiments we measured the performance after implementing the 3 main modifications and then we compared it to the original TIM. Then we evaluated the improvement achieved by tuning the learning rate. We evaluated the 3 first modifications together; however, further analysis should be done to fully understand individual and combined contributions.

4.2 Other approaches

As part of our work, we tried out different models from previous approaches for human motion prediction. In our experiments we observed that the performance of RNN used in [7] was surpassed by an adversarial training approach with the same model as a generator but using 2 RNN discriminators and a loss function that combines L1 distance loss with adversarial losses as in [8]. For that reason, we implemented more complex models using adversarial training.

One of the models that achieved a particularly good performance was Adversarial GCN (A-GCN) an approach similar to [6]. A-GCN consists of a GCN that learns the residuals between the target sequence and the original sequence with the last pose replicated 24 times. The model has 15 blocks of graph convolutional layers with two additional graph convolutional layers, one at the beginning and one at the end. Contrary to [6] we don't use DCT coefficients to represent the trajectory of joints but the original poses in rotation matrix representation and we use just the last 48 time-frames to predict the next 24 frames. The reasoning behind these two modifications is, firstly DCT coefficients encode the whole sequence in the frequency domain and our hypothesis was that just the immediate frames carry meaningful information to predict the next 24. And secondly, by selecting just a subset of the DCT coefficients we are losing part of the information. Since we are not using all frames, we want to use all the information we have available in that 48 time-frame sequence.

We didn't get better results than the model described in section 2. However, the performance was very close as shown in figure 4. For A-GCN the adversarial training didn't contribute as much as in the RNN model. Our hypothesis was that the discriminators we had were too simple and we needed a more complex model for them. Since the accuracy of both discriminators decayed very quickly to 50%, a more complex model would identify better the real and fake sequences, hence keep a good accuracy for longer and improve the performance of the generator as well. We implemented a GCN as the discriminator but we ended up with a very similar performance. Our plan for future work is to implement a different model for the discriminator and weight in a different way the L1 distance loss and adversarial loss.

5 CONCLUSION

In this work we introduce a new approach for human motion prediction that consists in modifications to GCN with TIM. Our experiments revealed that it is important to adjust the model depending on the length of the predicted sequence. Adding more different-sized 1D convolutional filters and changing the dilation and stride for the filters of the longer sub-sequences helps to detect higher-level patterns. It is enough to use just a few frames from the seed sequence to predict the next frames and this also allows us to increase the training set size. Alternative approaches like adversarial training are a potential way to improve the performance of the model but further exploration should be done in order to find the right model for the discriminators and right weights in the loss function.

REFERENCES

[1] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. 2015. Recurrent Network Models for Human Dynamics. arXiv:cs.CV/1508.00271

[2] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2013. 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 (2013), 1325–1339.

[3] Ashesh Jain, Amir Roshan Zamir, Silvio Savarese, and Ashutosh Saxena. 2015. Structural-RNN: Deep Learning on Spatio-Temporal Graphs. *CoRR* abs/1511.05298 (2015). arXiv:1511.05298 http://arxiv.org/abs/1511.05298

[4] Tim Lebailly, Sena Kiciroglu, Mathieu Salzmann, Pascal Fua, and Wei Wang. 2020. Motion Prediction Using Temporal Inception Module. In *Proceedings of the Asian Conference on Computer Vision*.

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

[6] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. 2019. Learning Trajectory Dependencies for Human Motion Prediction. *2019 Ieee/Cvf International*

Conference On Computer Vision (Iccv 2019) (2019), 9488–9496. https://doi.org/10.1109/ICCV.2019.00958

[7] Julieta Martinez, Michael J. Black, and Javier Romero. 2017. On Human Motion Prediction Using Recurrent Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4674–4683. https://doi.org/10.1109/CVPR.2017.497

[8] Yuxiong Wang, Liang-Yan Gui, Xiaodan Liang, and Jose M. F. Moura. 2018. Adversarial Geometry-Aware Human Motion Prediction. In *Proceedings of (ECCV) European Conference on Computer Vision*. Springer.