Machine Perception: Human Motion Prediction

Irfan Bunjaku
irfan.bunjaku@inf.ethz.ch

Laure Ciernik
lciernik@student.ethz.ch

Felix Sarnthein
safelix@student.ethz.ch

ABSTRACT

Human motion prediction aims to predict future body poses given observations of previous poses. In this report we extend existing methods and qualitatively compare them on the AMASS [4] dataset. We achieve the best results by using a graph convolutional network (GCN) preceded by an attention mechanism which we adapt from Mao et al. [7]. The attention mechanism allows to emphasize specific sub-sequences of motion given the current motion context, while the GCN captures spatial and temporal dependencies between joints. By extending the receptive field of the attention layer, we adapt the model to better suit the given data set and improve predictive performance.

1 INTRODUCTION

Many computer vision tasks seem very easy to humans and are performed on a daily basis but are extremely challenging for machines. This is no different for human motion prediction. Simple things such as handshakes would be impossible without humans natural ability to understand and predict the motion of other agents hands. The human mind has a natural gift to detect objects and even predict their projectile motion without knowing the physics underlying them. Human motion prediction has many applications in human-robot interaction, computer vision and autonomous driving. Especially robots that interact with humans either physically or virtually must be able to predict human movement. While long term human motion is hard to predict due to its high uncertainty and therefore with error accumulation, short term motion is largely determined by the physical properties of the current movement and has less uncertainty.

In this project we are concerned with short term motion prediction. We take an input sequence of 120 poses and predict the following 24 poses corresponding to 2 seconds and 400 milliseconds, respectively. There are multiple approaches to model this problem. On the one hand, one could consider a local scope and model the temporal dynamics to predict future frames based on previous frames. Alternatively one could fill in the unknown frames to construct an initial sequence of length 144 and consider the global scope by gradually transforming the entire sequence to an improved sequence where the last 24 frames contain the predicted poses.

We implement a human motion prediction model based on the work of Mao et al. [5, 7] and compare it to multiple other methods. As in [5] we represent the human motion in trajectory space instead of pose space by using the Discrete Cosine Transformation (DCT) to encode the temporal information.

We use a learnable graph convolutional networks to capture the spatial structure of the motion data, instead of using a predefined graph structure based on the kinematic chain. We additionally extend the attention based model proposed by Mao et al. [7] to use a variable size receptive field. The motion attention model exploits the historical information by dynamically adapting its focus on the previous motion to the current context. While Mao et al. use the exponential representation or 3D pose representation of the human joints, we follow Aksan et al. [1] and model the joint angles as local rotation matrices, with respect to the parent joint.

The attention mechanism allows us to emphasize different parts of past sequences which yields a significant improvement as our comparisons show.

2 RELATED WORK

Martinez et al. [6] introduce a sequence-to-sequence model, as is often used in natural language processing tasks, with a single layer of GRU cells. They apply a sampling based loss, using its own predictions during training. Further, they propose a residual architecture to model velocities instead of positions. This results in smoother short-term predictions. They give a good critical analysis of the current state of the art at the time and show that most methods are easily outperformed by a zero velocity approach which does nothing other than repeating the last frame of the input sequence.

Aksan et al. [1] further propose to decompose the prediction into individual joints by means of a structured prediction layer that explicitly models the joint dependencies. They implement a hierarchy of of small-sized two-layered linear neural network which are connected analogously to the kinematic chains in the human body.

Mao et al. [5, 7] also try to encode the spatial dependency of the human pose. Instead of using a predefined graph structure based on the kinematic chain, they treat the pose as a generic graph formed by links between every pair of joints and use a graph convolutional network (GCN) to learn the graph connectivity from data. Further, similarly to Martinez et al. [6] they do not predict the pose directly. Instead they represent the human motion in trajectory space by using the DCT (Discrete Cosine Transform) to encode the global temporal information. In a later work Mao et al. [7] introduce an attention layer to capture the similarity between the current motion context and the historical motion sub-sequences.

3 METHOD

In this section we explain the methods used. We start off by presenting the data used. Then we briefly introduce the investigated models and finally present our submitted method.

Data representation. For this project we use a subset of the AMASS [4] dataset. This is already separated into a training, test, and validation split. We use the SMPL [3] body-model where a human pose is represented by 15 joint angles. Each joint angle is represented by a 3 by 3 rotation matrix, given with respect to the parent angle. In total each pose is determined by $K = 135$ parameters. We also implemented an axis angle representation but this proved to be detrimental to performance.

Investigated Models. We investigate nine different models which we introduce ordered by increasing complexity. The first and most simple model is the Zero Velocity model as introduced by Martinez et al. [6] which does not try to model motion at all. It simply repeats the last input pose as prediction for the future 24 poses.

The next model uses a sequence to sequence architecture with a sampling-based loss proposed by Martinez et al. [6] . We consider different extensions and adjustments of this architecture, comparing GRU and LSTM cells and altering the number of cells used. We omit those that gave us no additional benefit for the sake of brevity and just state those that gave additional improvements. We evaluate three models based on the sequence to sequence architecture. One with one GRU cell, the next with two LSTM cells and the last with three LSTM cells.

We also use a vanilla deep recurrent neural network (RNN) as described by Aksan et al. [1], to model time-dependent human motion. We further try to enhance the predictive performance with a structural prediction layer (SPL) as described in [1], which decomposes the prediction into individual joints. This exploits the spatial dependency of the human body by a hierarchy of small-sized neural networks connected analogously to the kinematic chain.

3.1 Final Model

Now we come to the quintessence of our report and explain our final model. First we explain the attention layer used. Then we give the details of our prediction layer and explain the structure of the GCN used. Finally, we provide implementation details.

Attention Layer. We briefly discuss the attention layer. Given an input sequence of length N and wanting to predict the T following poses, we divide our input sequence into overlapping sub-sequences of size $M + T$, where T is the prediction window, in our case a sequence of length 24, and M is the receptive field, i.e the history based on which T is predicted. We use the first M poses as the key and the whole sub-sequence as the value of the attention mechanism. The query corresponds to the last M poses of an input sequence. Both the keys and the query use a simple convolutional layer as a flattening function to transform the representation of the pose sequences to vectors, we denote them by k_i and q, respectively.

As defined in [7] the attention scores are computed via the scaled dot-product between the query and the keys:

$$a_i = \frac{q \cdot k_i}{\sum_{j=1}^{N-M-T+1} q \cdot k_j} \tag{1}$$

The attention scores are then used to weight the DCT coefficient of each transformed sub-sequence.

In comparison to Mao et al. we aim to predict more poses per second and thus have a longer input sequence. This suggests that a larger receptive field M than used in [7] might be required. We adjust the method to use different sizes of M and investigate their benefit in our hyperparameter search.

The kernel size of the two convolutional layers in the attention mechanism depend on the receptive field. In contrast to Mao et al. [7] who use fixed kernel sizes of 6 and 5, we generalize their approach to variable kernel sizes depending on the size of the receptive field M. Let k_1 be the kernel size of the first convolutional layer and k_2 the kernel size of the second convolutional layer. We define them as follows.

$$k1 = \frac{M+T}{2} + 1 \qquad k2 = \frac{M+T}{2} \tag{2}$$

Prediction Layer. Similar to Mao et al. [7] we do not predict the poses directly, but encode the history in frequency space using DCT, followed by a GCN to encode the spatial and temporal connection. We use the full resolution of DCT coefficients which is equal to the sub-sequence length of $M + T$. As input for the GCN we use the concatenation of the weighted sum of the transformed sub-sequences as described previously, the transformation of the query and T copies of the last pose.

We use the same GCN with residual structure as described by Mao et al. [5, 7]. Our architecture uses 12 residual blocks, where each block contains two GCN layers and two additional layers. One of these is at the beginning to map the DCT coefficients to features and a final layer to decode the features to DCT residuals. The output contains predictions of the last M and the future T poses of the sequence in frequency space. We use the Inverse Discrete Cosine Transform (IDCT) to get the predicted poses, shown Figure 1. Each layer relies on a learnable weight matrix W and a learnable weighted adjacency matrix A which represents the strengths of the edges in the fully-connected graph which models the human body and a learnable weight matrix W. Hence every layer outputs a matrix of the form

$$H^{(p+1)} = \sigma(A^{(p)} H^{(p)} W^{(p)}) \tag{3}$$

where $\sigma(.)$ is an activation function.

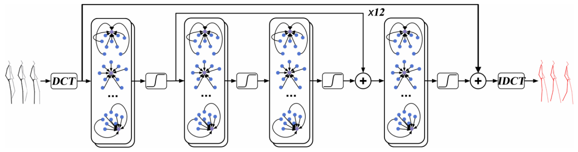

Figure 1: Illustration of the Network architecture used for the prediction layer taken from [5]

Implementation Details. We use a learnable weight matrix W of size 256 × 256. The learnable adjacency matrix A is of size 135 × 135, corresponding to the number of parameters in our pose representation. For the GCN we use a dropout value of 0.3 and *hyperbolic tangent* as activation function.

For the optimization we empirically determined to use the ADAM optimizer [2] with an initial learning rate of 0.005 and exponential weight decay of 0.98 for the learning rate every 330 optimization steps. As loss function we use the average l1 loss [5] defined as:

$$l = \frac{1}{K(M+N)} \sum_{t=1}^{M+T} \sum_{k=1}^{K} \left| \hat{x}_{t,k} - x_{t,k} \right| \tag{4}$$

To avoid exploding gradients we clip the gradient to a $\mathcal{L}_2$ norm of 1. Further we use a batch size of 128 to increase the stability of the gradient and trained the model for 1000 epochs.

4 EVALUATION

In this section we compare our method to several models of different complexity. To compare the results we use the mean joint angle difference between all joints, summed over all 24 target poses. Let R_{ij} be the rotation matrix for joint j in sequence i. Further, let $\hat{R}_{ij}$

be the prediction for R_{ij}, T the sequence length and J the number of joints, then the mean joint angle difference is defined as:

$$MJA = \frac{1}{TJ} \sum_{i=1}^{T} \sum_{j=1}^{J} \left\| \log \left(\hat{R}_{ij} \cdot R_{ij}^{T} \right) \right\|_2 \tag{5}$$

Table 1 shows the mean joint angle for each model evaluated on the validation set and on the public test set, i.e. the mean joint angle on the submission page. For all models, including the Zero Velocity model, we observe a higher mean joint angle of roughly 0.5 on the validation set compared to the evaluated predictions on the public score. Since the difference is consistent over all models we do not take this as a sign of overfitting to the public score.

We achieve the smallest deviation from the ground truth with a mean joint angle of 2.285 by using our final model (DCT and GCN with attention). There is a clear improvement over the GCN network without the attention mechanism which is clearly visible in Table 1 and Figure 2.

The sequence to sequence models (Seq2Seq) with one GRU cell and two LSTM cells are outperformed by the Zero Velocity model we used as a baseline. One reason for this might be the lack expressivity due to the simplicity of the encoder-decoder model. This would also explain the improved performance by adding more LSTM cells.

With the intent to introduce spatial dependencies, we tried to implement the SPL layer as described by Aksan et al. [1]. While the vanilla RNN model does impressively well, we see no benefit from the additional SPL layer. This might be due to an implementation error or due to a poor choice of hyperparameters. Due to the long training time of the SPL layer we forwent an extensive hyperparemeter search in favor of the GCN model. We also tried extending the vanilla RNN model similar to the sequence to sequence models. While adding additional LSTM cells helped for the sequence to sequence model, it just lead to overfitting for the vanilla RNN model.

The GCN models with DCT outperform all our previous RNN based methods. We attribute this to the encoding of the spatial dependencies and the predictions in the trajectory space.

Model	Validation	Public test
DCT and GCN with attention	**2.285**	**1.859**
DCT and GCN	2.529	2.076
RNN	2.742	2.318
RNN with SPL	2.879	2.317
Seq2Seq with 3 LSTM cells	3.733	3.376
Zero Velocity	4.175	3.616
Seq2Seq with 2 LSTM cells	4.220	3.714
Seq2Seq with 1 GRU cell	4.498	4.027

Table 1: Mean joint angle on validation set until 24th frame

Hyperparemeters. An important part of finding the best model is hyperparameter tuning. We already mentioned the parameters used. Here we will further evaluate their influence.

The batch size influences the accuracy of the estimated gradient. The larger the batch size the more accurate the gradient is. For all models except the GCN model we use a batch size of 16 which appeared to be a good choice, giving enough stability for the estimated gradient. We observed that increasing the batch size for those models did not improve performance. For the GCN model however we noticed different behaviour. Increasing the batch size to 128 improved the mean joint angle evaluated on the validation set, we assume due to increased stability in the gradient.

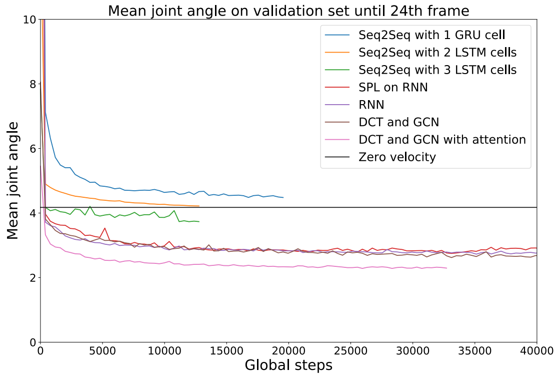

Figure 2: Mean joint angle on validation set until 24th frame. A global step corresponds to one optimization step.

Since Mao et al [7] predict less frames and have a shorter input sequence we decided to alter the receptive field M, i.e. the history depth that we use to predict the 24 target poses. Table 2 shows the mean joint angle evaluated on the validation set for different receptive field sizes. We conclude that 40 poses or about 0.67 seconds best describe sub-motions to identify the current motion context and predict the upcoming 24 poses.

Receptive field size	Mean joint angle
10	2.346
20	2.335
30	2.306
40	**2.285**
50	2.318
60	2.299
70	2.322
80	2.330
90	2.307

Table 2: Mean joint angle on the validation set until the 24th predicted frame for DCT and GCN with an attention mechanism model comparing different receptive field sizes.

5 CONCLUSION

Human motion prediction remains a hard task for machines as the Zero Velocity baseline impressively demonstrates. We extensively compared different methods and were able to slightly improve an existing state of the art architecture to do reasonable short term predictions. Future work could investigate different trajectory representations or imposing priors on potentially more complex graph structures.

6 SUPPLEMENTARY MATERIAL

The results of our hyper parameter search including all trained models are available here:
https://polybox.ethz.ch/index.php/s/uvGOCzBSufPRybm

REFERENCES

[1] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. 2019. Structured Prediction Helps 3D Human Motion Modelling. In *The IEEE International Conference on Computer Vision (ICCV)*. First two authors contributed equally.

[2] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:cs.LG/1412.6980

[3] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.

[4] 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.

[5] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. 2020. Learning Trajectory Dependencies for Human Motion Prediction. arXiv:cs.CV/1908.05436

[6] 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

[7] Mao Wei, Liu Miaomiao, and Salzemann Mathieu. 2020. History Repeats Itself: Human Motion Prediction via Motion Attention. In *ECCV*.