Lab CudaVision
Learning Vision Systems on Graphics Cards (MA-INF 4308)

CudaLab Project

13.07.2023

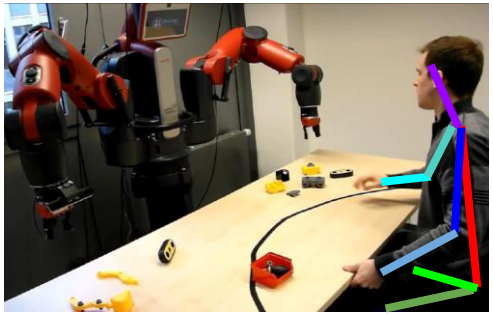
PROF. SVEN BEHNKE, SIMON BULTMANN

Contact: bultmann@ais.uni-bonn.de

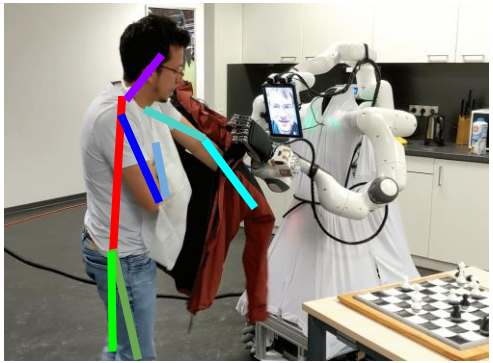
3D Human Pose Forecasting

Motivation

- Human-robot collaboration is a challenging task
  - Perception of the environment
  - Planning capabilities
  - ➢ **Predicting actions and behavior of nearby agents**

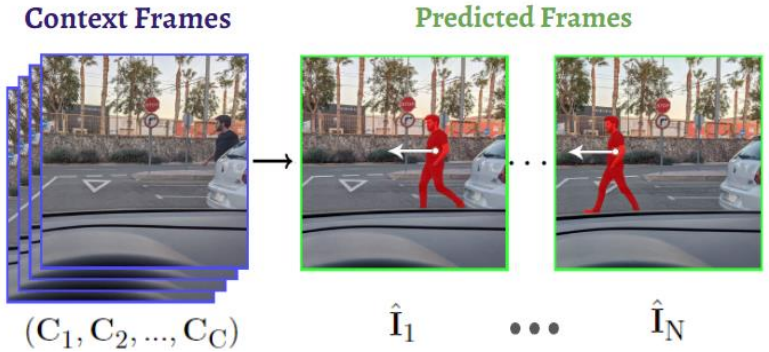


- Human pose is a good representation for:
  - Action recognition and prediction
  - Motion estimation
  - Planning and navigation



Human Pose Forecasting

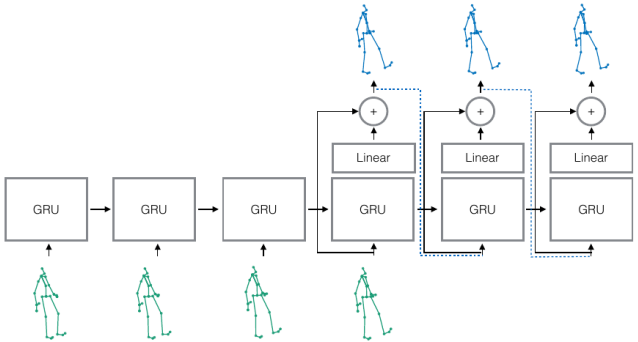
- Given a sequence of C seed poses, generate next N plausible poses
  - Predictions must be temporally consistent
  - Incorporate human motion dynamics
- Multiple applications: sports , anticipating human behavior, motion transfer, …



Related Work

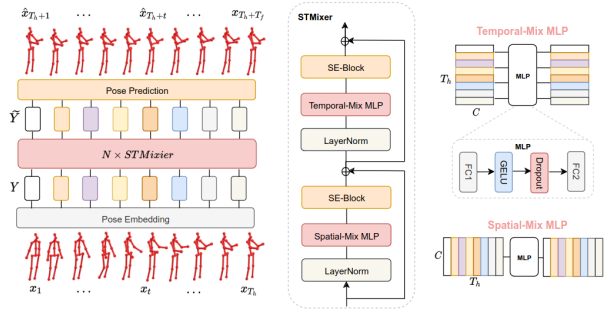
- Human motion prediction is an ongoing research topic (2015 - …)

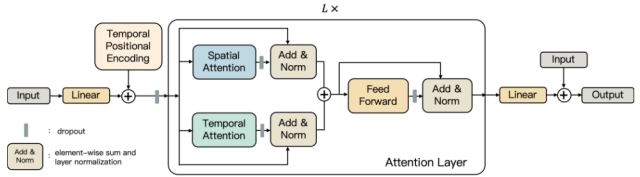
Recurrent neural networks



Graph neural networks, 3D-Convolutions, …

MLP- and Attention-based





Model

Model Inspiration

Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)

MotionMixer: MLP-based 3D Human Body Pose Forecasting

Arij Bouazizi[1,2*], **Adrian Holzbock**[2], **Ulrich Kressel**[1], **Klaus Dietmayer**[2] and **Vasileios Belagiannis**[3,†]
[1]Mercedes-Benz AG, Stuttgart, Germany
[2]Ulm University, Ulm, Germany
[3]Otto von Guericke University Magdeburg, Magdeburg, Germany
{arij.bouazizi, ulrich.kressel}@mercedes-benz.com, {adrian.holzbock, klaus.dietmayer}@uni-ulm.de, vasileios.belagiannis@ovgu.de

Abstract

In this work, we present *MotionMixer*, an efficient 3D human body pose forecasting model based solely on multi-layer perceptrons (MLPs). *MotionMixer* learns the spatial-temporal 3D body pose dependencies by sequentially mixing both modalities. Given a stacked sequence of 3D body poses, a spatial-MLP extracts fine-grained spatial dependencies of the body joints. The interaction of the body joints over time is then modelled by a temporal MLP. The spatial-temporal mixed features are finally aggregated and decoded to obtain the future motion. To calibrate the influence of each time step in the pose sequence, we make use of squeeze-and-excitation (SE) blocks. We evaluate our approach on Human3.6M, AMASS, and 3DPW datasets using the standard evaluation protocols. For all evaluations, we demonstrate state-of-the-art performance, while having a model with a smaller number of parameters. Our code is available at: https://github.com/MotionMLP/MotionMixer.

1 Introduction

Forecasting 3D human motion is at the core of many different applications ranging from virtual reality to autonomous driving [Wiederer *et al.*, 2020] and robotics [Gui *et al.*, 2018]. Fundamentally, the task of human motion prediction is defined as the prediction of future body poses from past ones. To model the human spatial-temporal dynamics, classic approaches adopted Hidden Markov Models [Kulić *et al.*, 2012] or Gaussian Processes [Wang *et al.*, 2007]. Despite the tangible progress in predicting periodic motion, these approaches impose strong assumptions on body pose, leading to performance degradation. Furthermore, it is difficult for these methods to forecast reliable 3D poses because of the complicated bio-mechanical kinematics. Even manually imposing expert knowledge does not help prior-based approaches to generalise to new environments and subjects.

Figure 1: Long-term predictions of *MotionMixer* for the actions *Directions* and *Posing* of the Human3.6M dataset. The first and the third line indicate the ground-truth 3D human motion. The frames on the left are the observations. The right part, shown in pink is the long-term motion prediction. One every three frames are shown. The model predictions accurately match the ground-truth body poses.

Recently, the availability of large-scale datasets, e.g. Human3.6M [Ionescu *et al.*, 2013], AMASS [Mahmood *et al.*, 2019] or 3DPW [von Marcard *et al.*, 2018], the development of human pose estimation algorithms [Belagiannis *et al.*, 2014; Bouazizi *et al.*, 2021] and the advent of deep learning methods pushed the evolution towards forecasting future 3D poses with less priors. Several learning-based approaches were proposed to tackle the problem of 3D human motion prediction. Methods like [Fragkiadaki *et al.*, 2015; Martinez *et al.*, 2017; Tang *et al.*, 2018] build upon the success of recurrent neural networks (RNNs) to better model the temporal correlation between the human body joints. Nev-

[*]Contact Author
[†]Most of this work was done while Vasileios Belagiannis was with Ulm University.

A Spatio-temporal Transformer for 3D Human Motion Prediction

Emre Aksan[1] Manuel Kaufmann[1] Peng Cao[2*] Otmar Hilliges[1]
[1]ETH Zürich, Department of Computer Science [2]Massachusetts Institute of Technology
[1]{eaksan,kamanuel,otmarh}@inf.ethz.ch [2]pengcao@mit.edu

Abstract

We propose a novel Transformer-based architecture for the task of generative modelling of 3D human motion. Previous work commonly relies on RNN-based models considering shorter forecast horizons reaching a stationary and often implausible state quickly. Recent studies show that implicit temporal representations in the frequency domain are also effective in making predictions for a predetermined horizon. Our focus lies on learning spatio-temporal representations autoregressively and hence generation of plausible future developments over both short and long term. The proposed model learns high dimensional embeddings for skeletal joints and how to compose a temporally coherent pose via a decoupled temporal *and* spatial *self-attention mechanism. Our dual attention concept allows the model to access current and past information directly and to capture both the structural and the temporal dependencies explicitly. We show empirically that this effectively learns the underlying motion dynamics and reduces error accumulation over time observed in auto-regressive models. Our model is able to make accurate short-term predictions and generate plausible motion sequences over long horizons. We make our code publicly available at* https://github.com/eth-ait/motion-transformer.

1. Introduction

3D human motion modelling is typically formulated as the prediction of future poses given a past horizon. Humans are able to effortlessly forecast the complex dynamics of motion in a plausible fashion due to our strong structural and temporal priors. From a learning perspective this problem can be seen as a generative modelling task: A network learns to synthesize a sequence of human poses, where the model is conditioned on the seed sequence. The task requires learning of pose priors for natural articulation and of underlying dynamics to yield plausible motion predictions. Since these factors are highly latent and entangled, introducing inductive biases and tailoring architectures for the task is essential for modelling of 3D human motion data.

Given the temporal nature of human motion, it is not surprising that recurrent neural networks (RNNs) are the most popular choice [2, 11, 20, 27, 31, 37]. RNNs model short and long-term dependencies by propagating information through their hidden state. Convolutional neural networks (CNN) in a sequence-to-sequence framework have also been proposed [13, 21, 23]. Such approaches focus on modelling the temporal aspect of the problem following an auto-regressive approach, but neglect structural priors. Instead, vectorized poses are passed as inputs at every step and the spatial dependencies are assumed to be learned implicitly. However, considering the skeletal structure in the architectural level is shown to be an effective inductive bias in [2, 4, 20, 24].

Since the auto-regressive approach factorizes the predictions into step-wise conditionals based on previous predictions, these models tend to accumulate error over time and eventually the predictions collapse to a non-plausible pose. This issue can be associated with the exposure bias problem [32] due to discrepancies between data and model distributions. Previous work has applied various strategies to work around this problem, such as using model predictions during training [27, 31], applying noise to the inputs [2, 12, 20, 23], or using adversarial losses [23, 37].

Recent works [6, 26, 38] model the temporal aspects of 3D human motion by encoding every joint's trajectory with the discrete cosine transformation (DCT). Both the observations and the predicted future frames are represented as a set of DCT coefficients which are then used to model inter-joint dependencies. Such an implicit modelling of the temporal information inherently mitigates the failure cases of the auto-regressive models. DCT appears to be an effective non-learning based representation.

In this work, we present a novel architecture for 3D human motion modelling, which attempts to learn a spatio-temporal representation explicitly without relying on the propagation of a hidden state as in RNNs or fixed temporal encodings such as DCT coefficients. Our approach is motivated by the recent success of the Transformer model [35]

[*]Affiliated with Peking University at the time of work.

On human motion prediction using recurrent neural networks

Julieta Martinez[*1], Michael J. Black[2], and Javier Romero[3]
[1]University of British Columbia, Vancouver, Canada
[2]MPI for Intelligent Systems, Tübingen, Germany
[3]Body Labs Inc., New York, NY
julm@cs.ubc.ca, black@tuebingen.mpg.de, javier.romero@bodylabs.com

Abstract

Human motion modelling is a classical problem at the intersection of graphics and computer vision, with applications spanning human-computer interaction, motion synthesis, and motion prediction for virtual and augmented reality. Following the success of deep learning methods in several computer vision tasks, recent work has focused on using deep recurrent neural networks (RNNs) to model human motion, with the goal of learning time-dependent representations that perform tasks such as short-term motion prediction and long-term human motion synthesis. We examine recent work, with a focus on the evaluation methodologies commonly used in the literature, and show that, surprisingly, state-of-the-art performance can be achieved by a simple baseline that does not attempt to model motion at all. We investigate this result, and analyze recent RNN methods by looking at the architectures, loss functions, and training procedures used in state-of-the-art approaches. We propose three changes to the standard RNN models typically used for human motion, which result in a simple and scalable RNN architecture that obtains state-of-the-art performance on human motion prediction.

1. Introduction

An important component of our capacity to interact with the world resides in the ability to predict its evolution over time. Handing over an object to another person, playing sports, or simply walking in a crowded street would be extremely challenging tasks without our understanding of how people move, and our ability to predict what they are likely to do in the following instants. Similarly, machines that are able to perceive and interact with moving people, either in physical or virtual environments, must have a notion of how people move. Since human motion is the result of both physical limitations (*e.g.* torque exerted by muscles, gravity, moment preservation) and the intentions of subjects (*how* to perform an intentional motion), motion modeling is a complex task that should be ideally learned from observations.

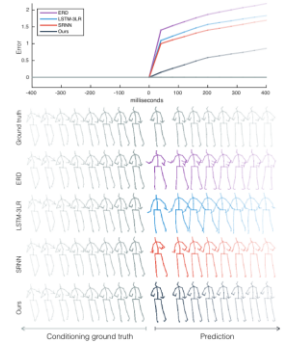


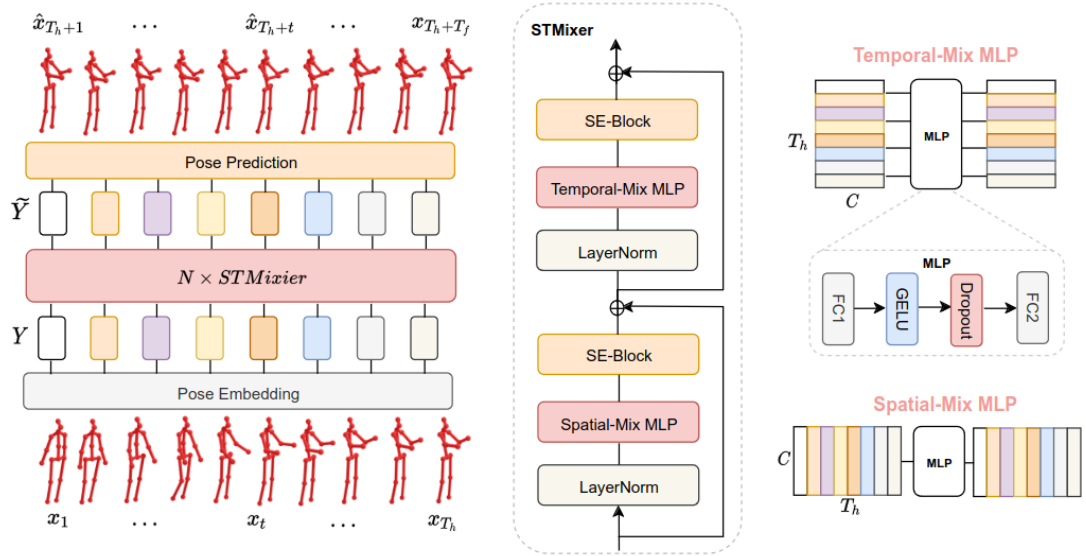
Figure 1. Top: Mean average prediction error for different motion prediction methods. Bottom: Ground truth passed to the network is shown in grey, and short-term motion predictions are shown in colour. Previous work, based on deep RNNs, produces strong discontinuities at the start of the prediction (middle column). Our method produces smooth, low-error predictions.

Our focus in this paper is to learn models of human motion from motion capture (mocap) data. More specifically, we are interested in human motion prediction, where we forecast the most likely future 3D poses of a person given their past motion. This problem has received interest in a

[*]Research carried out while Julieta was an intern at MPI.

Proposed Model (1)

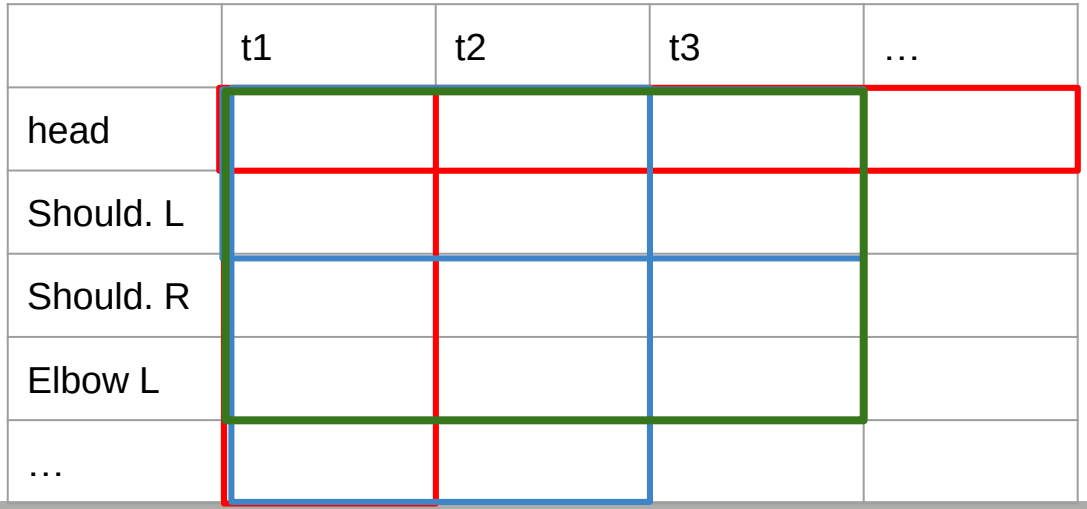
- MLP-based, spatio-temporal mixing[1]



1: Bouazizi, Arij, et al. "MotionMixer: MLP-based 3d human body pose forecasting." International Joint Conference on Artificial Intelligence, (IJCAI). 2022.

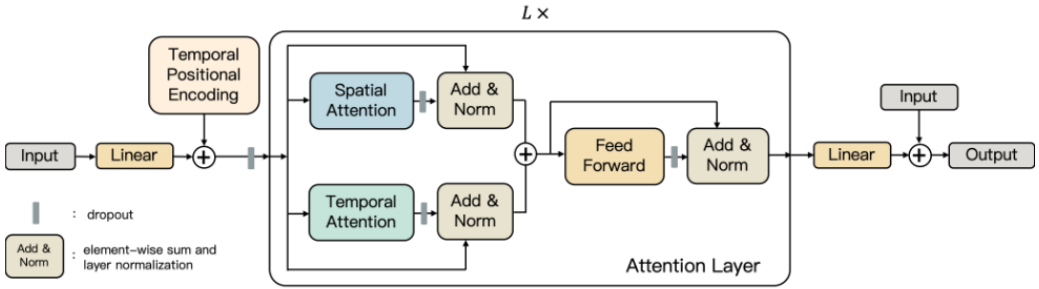
Proposed Model (1)

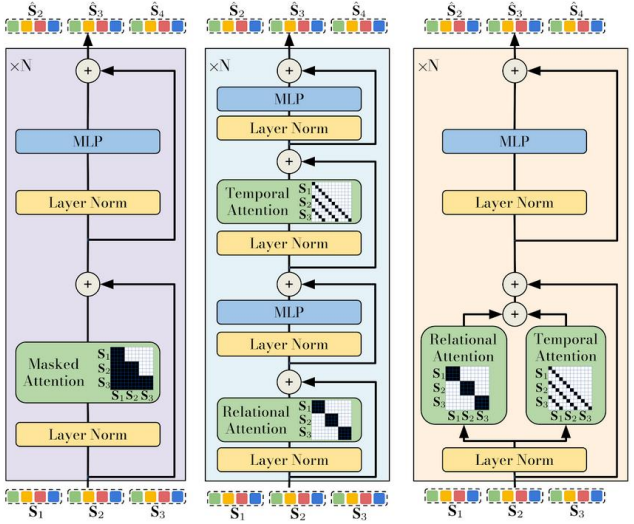
- Data is processed in temporal and joint-dimensions in alternating manner
- Implement with 1D Convolutions (instead of MLP)
- Extend to rectangular Convolutional kernels
- Compare also with standard, square convolutional kernels



Proposed Model (2)

- Attention-based, spatio-temporal transformer[2]





[https://sites.google.com/view/ocvp-vp]

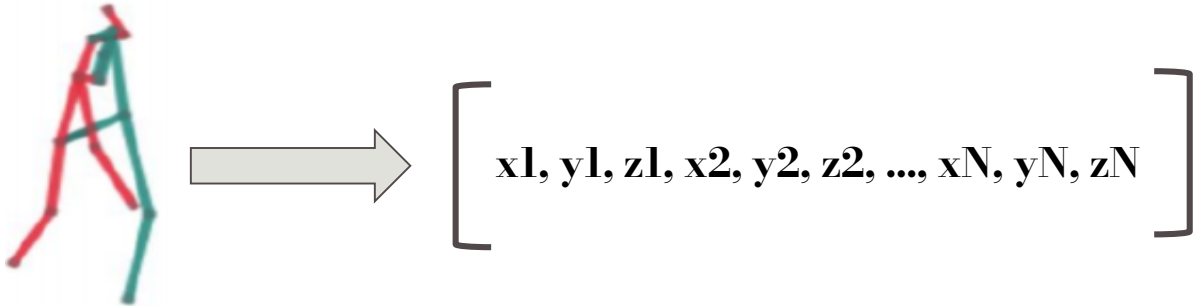
- Compare with **sequential** spatio-temporal processing and standard transformer

2. Aksan, Emre, et al. "A spatio-temporal transformer for 3d human motion prediction." 2021 International Conference on 3D Vision (3DV). IEEE, 2021.

Pose Representation

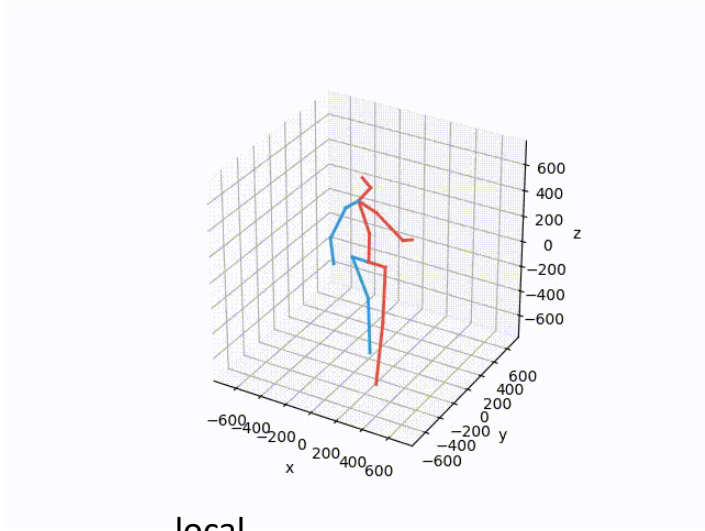
- Pose is parameterized as list of coordinates or joint angles
  - One dimensional representation
  - (x,y,z): Shape is 3N, where N is the number of joints
  - (joint angles): Different angle representations (quaternions, rot. mat, axis-angle)
    - Re-normalization /-orthogonalization may be needed

- Pose can be preprocessed:
  a. Normalize each coordinate by the maximum x, y & z values respectively

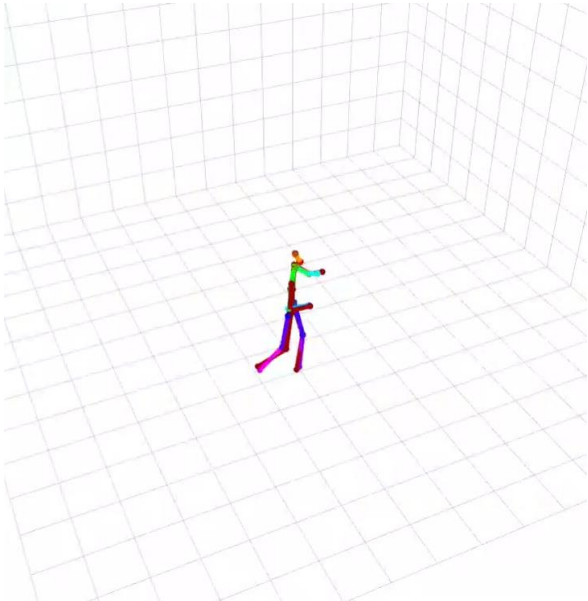


Pose Representation

- Global vs. local movement
  - Local vs. global movement (root joint subtracted or not)



local



global

Model

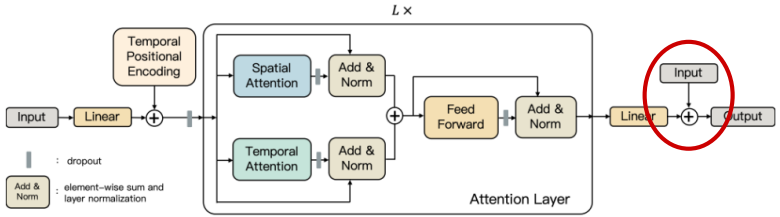
- Encoder:
  - Maps input poses into higher-dimensional representation
  - One single fully connected layer might suffice

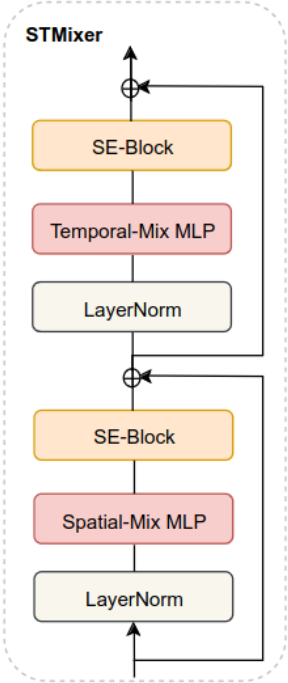
- Pose Predictor:
  1. Spatio-Temporal Mixer: Implement as 1D Convolution
     - Compare with rectangular and standard Convolutions
  2. Attention-based with spatio-temporal attention
     - Compare parallel and sequential spatio-temporal attention and vanilla transformer
- Decoder:
  - Maps output of predictor back to pose space
  - One single fully connected layer might suffice

Residual Connection

- Poses cannot change much between two consecutive time steps

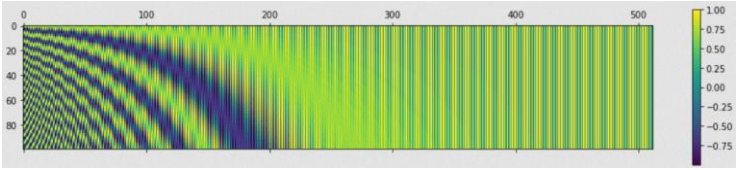
- Residual connection between time steps
  - Network must only model changes
  - Faster learning



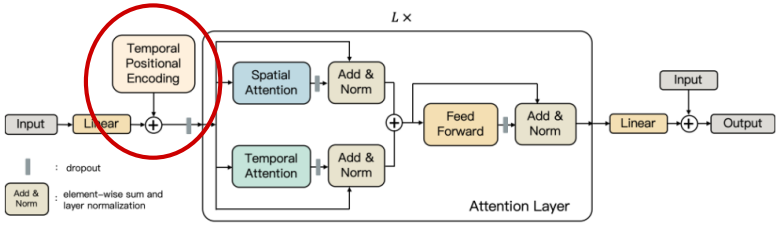


Positional Encoding

- Transformer is permutation agnostic w.r.t. inputs
- We need positional encoding to know the sequence order
  - Index is mapped to sin() / cos() vector
  - Embedding and pos. encoding vector are added
- Consider separate channels for timestep and joint idx., respectively



[https://machinelearningmastery.com/a-gentle-introduction-to-positional-encoding-in-transformer-models-part-1/]

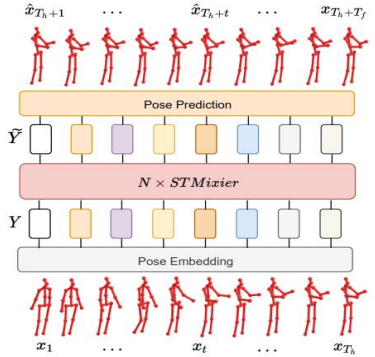


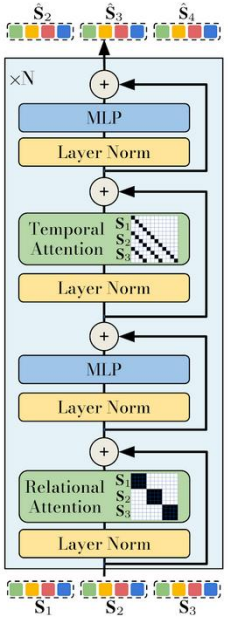
Joint seq.	H_1	N_1	RS_1	LS_1	H_2	N_2	RS_2	LS_3	…
Pos. enc. (1)	0	1	2	3	4	5	6	7	…
Pos. enc. (2)	0+0	0+1	0+2	0+3	1+0	1+1	1+2	1+3	…

Model Flow

Autoregressive model:

- Sequence to sequence prediction using sliding window
  - Sequence of Ns + Np frames: Ns seed-frames, Np predictions.
    - Frames [0, … Ns-1] -> predict Frame Ns
    - Frames [1, … Ns] -> predict Frame Ns+1
    - …
    - Frames [Np-1, … Ns+Np-2] -> predict Frame Ns+Np-1
- Predictions are re-encoded and used as inputs

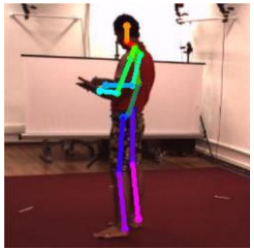


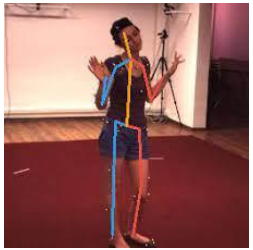


Datasets

Human 3.6M Dataset

- Dataset used as a benchmark for many tasks
  - Pose estimation and forecasting
  - Video prediction
- Stats and characteristics
  - 3.6 million 3D human poses and images
  - 11 professional actors (6 male, 5 female)
  - 17 scenarios (discussion, smoking, ...)
- Data is available at:
  - https://github.com/enriccorona/human-motion-prediction-pytorch
  - Download-link for pre-processed data in axis-angle representation (called “exp. map”)
  - Helper tools for forward kinematics, etc.

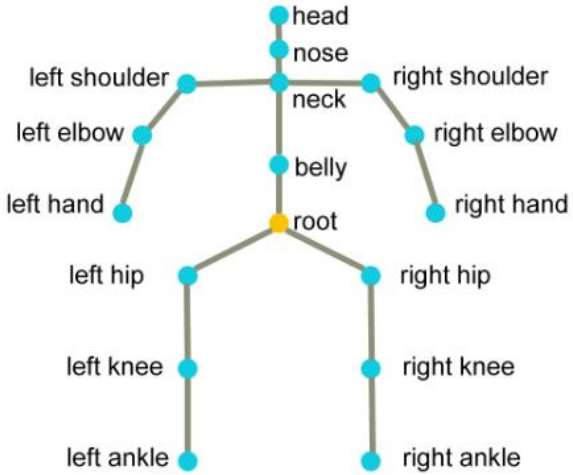




Skeleton Model

- Originally, 32 joints are in the dataset
- Most often, 17 joints are used
- Motion Mixer uses 22 (20) joints

- Toolbox (orig. data):
  https://github.com/CHUNYUWANG/H36M-Toolbox
- Forward kinematics (axis-angle data):
  https://github.com/MotionMLP/MotionMixer/blob/main/utils/forward_kinematics.py

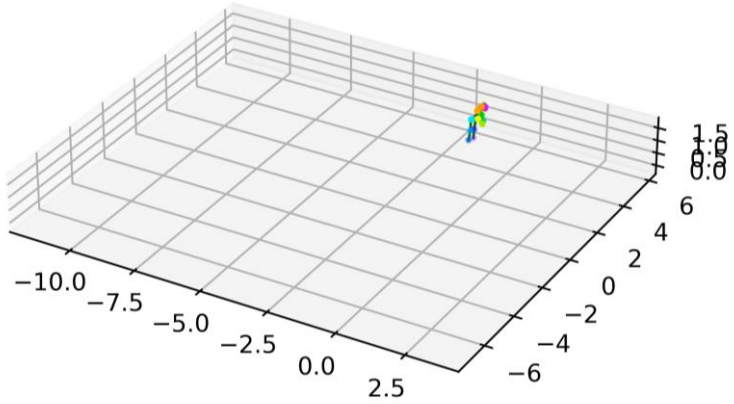


joint_name = ["Hips", "RightUpLeg", "RightLeg", "RightFoot", "RightToeBase", "Site", "LeftUpLeg", "LeftLeg", "LeftFoot", "LeftToeBase", "Site", "Spine", "Spine1", "Neck", "Head", "Site", "LeftShoulder", "LeftArm", "LeftForeArm","LeftHand", "LeftHandThumb", "Site", "L_Wrist_End", "Site", "RightShoulder", "RightArm","RightForeArm", "RightHand", "RightHandThumb", "Site", "R_Wrist_End", "Site"]

joint_idx = [0, 1, 2, 3, 6, 7, 8, 12, 16, 14, 15, 17, 18, 19, 25, 26, 27]

Additional Test Data

- Test generalization on sequences captured in our lab
- Data available here:
  https://cloud.vi.cs.uni-bonn.de/index.php/s/jff3KRejPLpLmS9
- Data in .json format and a sample script for visualization
- Slightly different skeleton model
  - (no ‘belly’ or ‘head’, but eyes and ears)



Training & Evaluation

Training and Prediction

- Datasets:
  - Train & evaluate on Humans3.6M, using different pose representations
  - Use the official train-test splits
  - Use a downsampling factor of 2 (50Hz -> 25Hz (1 / 40ms))
    - [f0, f1, f2, f3, …, f38, f39] → [f0, f2, …, f38]

- Train and evaluate with sequence of size:
  - For joint coordinates: (Batch_size, 20, 3 * N_jnts)
  - For joint angles: (Batch_size, 20, K * N_jnts), K depends on angle representation
- Use 10 frames as seed frames, and train to predict the next 10 frames.
  (400ms seed, 400ms prediction)

Training and Prediction

- Model:
  - Train & evaluate on **one** of the following:
    - **MLP-based** with different convolutional kernel design
    - **Attention-based** with different spatio-temporal transformer modules
  - Choose your design options: residual connections, pos. encoding, …
  - Teacher forcing vs. no teacher forcing
    - using ground-truth or previous prediction as input for autoregressive processing
- Criterion:
  - Use L2 loss for joint coordinates (MPJPE) or angle representations
  - (Optional) Add an additional adversarial loss

Evaluation

- Measure performance on the 10 predicted frames
  - Also evaluate for longer predictions (e.g. 25 frames = 1 second prediction)
- Evaluate using the following metrics[5]:
  - Euler Angle
  - Geodesic (joint angle)
  - **MPJPE** (position)
  - PCK (area-under-the-curve (AUC) using thresholds $0.001 \leq \rho \leq 0.30$)
- Qualitative evaluation by observing predicted frames

Aksan, Emre, Manuel Kaufmann, and Otmar Hilliges. "Structured prediction helps 3d human motion modelling." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.
https://arxiv.org/abs/1910.09070
https://github.com/eth-ait/motion-transformer/blob/master/metrics/motion_metrics.py
https://github.com/eth-ait/motion-transformer/blob/dd2c85872cb48a6ffab85b50e3a8342f401c6ddd/common/constants.py#L102

Project Goals and Deliverables

Passing Requirements

1. Implement one of the model architectures, pipelines and utils
2. Train your models to achieve best possible results on Human3.6M
   - You must implement and train the described model with different design options
   - Make changes and train further models to achieve better results
   - Evaluate on additional test data from real-world setup to show generalization
3. Create overview notebook
4. Write project report

Deliverables

- Complete codebase
  - Clean and structured
  - Not just a notebook!
- Trained model checkpoint and (tensorboard, WandB, …) logs
- Overview notebook (.ipynb & .html) showing main functionalities:
  - Load data
  - Load pretrained model
  - Display some results
- Project report

Grading

- Results and Experiments **55%-60%:**
  - Performing several experiments and obtaining good results
  - Additional experiments: ablation study, changes in the model, …
- Codebase & Overview Notebook **20%:**
  - Implement all functionalities
  - Modularity and structure
- Report **20%-25%**

Project Report

- Document your work in the project report
- Try to be brief, but readable and informative
- Include figures and tables
- Use *BibTex* for the references
- I expect 6-12 pages, but highly depends on number and size of imgs/tables
- Use the following template
  - https://www.overleaf.com/read/tmnvhrsdmjrp

Additional Experiment Ideas

- Try your own ideas!
- Investigate data preprocessing
- Investigate different pose representations
- Tweak the model
  - Change modules (num. layers, num. kernels, ...)
  - Investigate different predictors (Convolution kernel shapes, Transformer)
- Investigate different training strategies or transfer learning:
  - Use additional loss functions
  - Use adversarial supervision (add Discriminator)
- Make changes to the model
  - Stochastic model: https://arxiv.org/abs/1802.07687

Important Dates

- **13.07**: Starting date
- **29.08-09.09**: Revision session (reach out to me for an appointment!) (usually a remote meeting)
- **15.09**: Draft submission due (optional, if you want some feedback)
- **30.09**: Final submission due

Questions?

References

1. Bouazizi, Arij, et al. "MotionMixer: MLP-based 3d human body pose forecasting." International Joint Conference on Artificial Intelligence, (IJCAI). 2022.
2. Aksan, Emre, et al. "A spatio-temporal transformer for 3d human motion prediction." 2021 International Conference on 3D Vision (3DV). IEEE, 2021.
3. Villar-Corrales, Angel et al. “Object-Centric Video Prediction via Decoupling of Object Dynamics and Interactions.” ArXiv abs/2302.11850 (2023).
4. Karapetyan, Villar-Corrales et al. "Video Prediction at Multiple Scales with Hierarchical Recurrent Networks." arXiv preprint arXiv:2203.09303 (2022).
5. Martinez, Julieta, Michael J. Black, and Javier Romero. "On human motion prediction using recurrent neural networks." IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.
6. Aksan, Emre, Manuel Kaufmann, and Otmar Hilliges. "Structured prediction helps 3d human motion modelling." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.
7. Catalin Ionescu, et al., “Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments”, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI). 2014