3D Pose Estimation with Joint Group Spatial-Temporal Transformer

by

Yichen Yang

A thesis submitted to the
Department of Computer Science
in conformity with the requirements for
the degree of Master of Science

Bishop's University
Canada
June 2023

Copyright © Yichen Yang, 2023

Abstract

Transformer-based approaches have led to significant improvements in 3D human pose estimation (HPE) from 2D pose sequences, achieving state-of-the-art (SOTA) performance. However, current SOTAs fall short of capturing the spatial-temporal correlations of joints at different levels simultaneously. In this thesis, we present a joint group spatial-temporal transformer (JSTFormer). This transformer consists of three types of transformer encoders, a fusion module, a regression head, and a center frame extraction module to get the temporal-spatial correlation at different levels and further refine the transformer's output. We divide the human joints into three joint groups based on pose grammar. Extensive experiments on two datasets (Human3.6M and MPI-INF-3DHP) demonstrate that our work achieves competitive performance on benchmarks.

Acknowledgements

I would like to thank the Computer Science department at Bishop's University for giving me the opportunity to pursue a Master's degree. I would like to underline the support, patience, and guidance received from Dr. Russell Bulter, without him, none of this would have been possible. I would like to thank all other professors in the Department of Computer Science at Bishop's University, Dr. Stephen Bruda, Dr. Lin Jensen, Dr. Madjid Allili, Dr. Mohammed Ayoub Aloui Mhamd, and Dr. Bentabet from whom I learned a lot. I am sincerely grateful to Bishop's University Foundation and Dr. Russell Bulter for awarding me the scholarship. The financial support provided will greatly alleviate the financial burden of my studies.

Contents

Chapter 1

Introduction

Monocular 3D human pose estimation is a subfield of computer vision that estimates the 3D pose of a human body using a single 2D image or video frame as input. It is applied to a wide range of applications (e.g. motion capture[33], human-computer interaction[29], sports analysis[1], and medical diagnosis[39]). There are two main categories of 3D human pose estimation: direct estimation and 2D-to-3D lifting approaches. Direct estimation methods focus on directly estimating the position of body joints in 3D space from 2d images or video frames without intermediately estimating the 2D pose representation. In contrast, 2D-to-3D lifting approaches aim to reconstruct the 3D human pose from an intermediately estimated 2D pose. It has become popular in 3D human pose estimation, mainly due to the excellent performance of state-of-the-art 2D pose detectors and more accurate 3d pose estimation results. The success of 2D-to-3D lifting methods can be attributed to the robustness of 2D pose detectors and the efficiency of lifting algorithms in learning complex mapping from 2D joint positions to 3D space.

However, mapping 2D poses to 3D poses is a non-trivial task. Various potential 3D poses could be generated from the same 2D pose due to depth ambiguity and occlusion[44]. A single 2D pose can correspond to multiple 3D poses because the third dimension (depth) information is lost during the 2D projection, which leads to an ill-posed problem when trying to recover the 3D pose from 2D key points. To alleviate this issue, researchers have proposed different approaches to lifting 2D poses to 3D, including optimization-based methods, geometric constraints, and deep learning-based methods[[19], [19], [30], [18], [2]]. Several methods exploit additional information, such as temporal consistency in video sequences or multiple views, to improve 3D pose estimation. For example: [30] proposes a method that incorporates temporal convolutions in a CNN architecture for 3D pose estimation from 2D key points. It leverages temporal information by utilizing semi-supervised training with both labeled and unlabeled video data to improve the model's performance. [13] proposes a robust multi-view 3D human pose estimation framework that effectively utilizes information from multiple camera views to overcome these

challenges and improve the accuracy of the estimated poses. [11] proposes a differentiable pose augmentation framework specifically designed for 3D human pose estimation to enhance the robustness and generalization capabilities of the pose estimation model.

On the one hand, CNN-based approaches that rely on dilation techniques can have limited temporal connectivity, making it difficult to capture long-range dependencies in the data. On the other hand, Recurrent networks[16] can model sequential correlation, but they often struggle to capture complex temporal dynamics over long sequences due to their inherent sequential nature. The Transformer is a deep learning model architecture introduced by [35], which was originally designed for natural language processing (NLP) tasks such as machine translation but has since been applied to a wide range of applications, including computer vision[12], speech recognition[7], and reinforcement learning[4]. Moreover, Transformers can process input sequences of varying lengths and capture complex long-range dependencies, which is beneficial for many temporal tasks, such as video understanding[10], time-series analysis[34], machine translation[38]. Thanks to the self-attention mechanism of the transformer, relationships between joints in a human body and contextual information from the input data can be distinctly captured, and The transformer architecture has shown great potential for learning temporal representations across frames in sequences. Several notable works, including [21], [17], [24], and [47] have showcased the remarkable potential of transformer architectures in 3D human pose estimation tasks. These approaches effectively capture complex spatial-temporal correlations, enabling them to achieve state-of-the-art performance in their respective domains. Concurrently, pose grammar[9] has been increasingly employed to facilitate the process of 3D pose estimation tasks more sophisticatedly. The HSTFormer[31] uses an innovative transformer-based framework designed to structurally model multiple levels of joint spatial-temporal correlations in a bottom-up fashion. This approach emphasizes the importance of capturing hierarchical relationships for more accurate 3D pose estimation. Inspired by pose grammar and the vision transformer architecture, this study integrates a temporal-spatial transformer encoder and employs a joint group transformer encoder to accurately predict 3D human poses in a more formal and sophisticated manner.

Chapter 2

Related Work

Direct 3D Pose Estimation and 2D-to-3D Lifting are two different approaches to estimating 3D human poses from 2D images or video sequences. At the early stage, Direct 3D Pose Estimation is widely adopted. However, Direct 3D pose estimation generally requires more complex models and training procedures than the 2D-to-3D lifting approach, making it more difficult to develop, implement, and optimize.

2.0.1 2D-to-3D Lifting

The 2D-to-3D Lifting approach involves a two-step process. First, 2D human poses are estimated from the input images or video frames. Second, the 2D poses are ”lifted” or converted into 3D poses. [3] present an simple approach to 3D human pose estimation by performing 2D pose estimation, followed by 3D exemplar matching. [27] proposed a simple and effective approach to estimate 3D joint locations using a fully connected residual network based on 2D joint locations from a single frame. However, using videos instead of one single image for 3D human pose estimation offers several advantages due to the availability of temporal information. The continuous frames in a video provide context about the motion and enable the model to capture the dynamics of human movements more effectively. [14] proposed a method for 3D human pose estimation that leverages temporal information by using a recurrent neural network (RNN) with Long Short-Term Memory (LSTM) cells. But many of the earlier works in 3D pose estimation [[40], [37], [23], [20]] do not explicitly consider the kinematic correlations of human joints when projecting the joint coordinates to a latent space.

2.0.2 Pose Grammar

Pose grammar is a useful approach to modeling the kinematic correlations of human joints in 3D human pose estimation.

The idea behind pose grammar is to represent the human body as a hierarchical structure with rules that govern the possible configurations and relationships

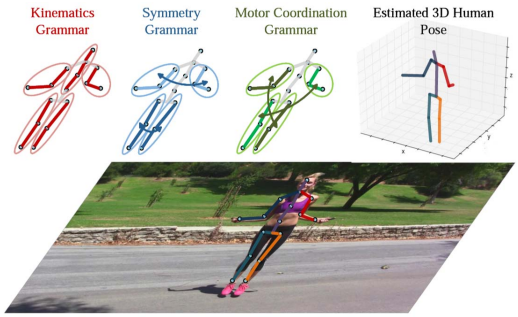

Figure 2.1: Illustration of human pose grammar, which expresses the knowledge of human body configuration. This figure is from [9]. Inspired by [9], we divide the human body into three parts based on motor coordination grammar and feed the three parts into our joint group transformer encoder.

between the joints. By incorporating these rules into the estimation process, a pose grammar-based method can effectively enforce kinematic constraints and predict plausible human poses. [9] proposed a deep grammar network, which consists of a base network that efficiently captures pose-aligned features and a hierarchy of Bidirectional RNNs (BRNN) on the top to incorporate a set of knowledge regarding human body configuration explicitly. It considers three kinds of human body dependencies and relations which are kinematics grammar, symmetry grammar, and motor coordination grammar. [31] utilized transformer architecture and human pose grammar to capture short-range and long-range dependencies in human pose sequences. It processes pose sequence in a hierarchical paradigm, from joints to body parts, and eventually to the entire pose in which four transformer encoders are concatenated following the kinesiological orders: spatial transformer encoder, joint temporal transformer encoder, body-part temporal transformer encoder, and pose temporal transformer encoder.

Inspired by their works, we group the human joints into three groups based on motor coordination grammar in Figure 2.1. Subsequently, we employ vanilla transformer encoders to capture the spatial-temporal relationships between joint, and one joint group transformer encoder to get the spatial-temporal correlation between joint groups based on motor coordination grammar. In the joint group transformer encoder, we assign three joint groups to the query, key, and value components in the multi-head cross-attention mechanism to predict the correlation between each group. In our proposed joint group transformer encoder architecture, the transformer incorporates position embeddings and multi-head cross-attention. It includes node feature normalization, a feed-forward aggregator for attention head outputs, and residual connections. These components enable the model to scale effectively with stacked layers.

2.0.3 Vision Transformers

Recently, there has been an emerging interest in applying transformers to vision tasks.

Vision Transformer (ViT) was introduced in the paper [8]. It demonstrates that the Transformer architecture, which was initially designed for natural language processing tasks, could also be effectively applied to computer vision tasks, particularly image recognition. ViT is a class of models that apply the Transformer architecture, originally designed for natural language processing tasks, to computer vision problems.

In the domain of 3D human pose estimation, ViT has been extensively employed and has demonstrated promising results. [47] designs a spatial-temporal transformer structure to comprehensively model the human joint relations within each frame and the temporal correlations across frames, then outputs an accurate 3D human pose of the center frame. [21] employs a triple attention mechanism to merge multiple hypotheses into a unified representation effectively. By using three attentions in a transformer, Multi-Hypothesis Transformer for 3D Human Pose Estimation Model merges multiple hypotheses into a single converged representation and then partitions it into several diverged hypotheses and therefore learns spatial-temporal representations of multiple plausible pose hypotheses. Subsequently, it partitions this representation into various diverged hypotheses, enabling the model to learn spatial-temporal representations of multiple plausible pose hypotheses in a sophisticated manner. [43] employs a distinct temporal transformer block to individually model each joint's temporal motion and a separate spatial transformer block to capture inter-joint spatial correlations. By alternating these two blocks, the model achieves superior spatial-temporal feature encoding. Consequently, [43] surpasses the state-of-the-art approach with a significant improvement of 10.9% in P-MPJPE and 7.6% in MPJPE metrics.

Chapter 3

Method

In this thesis, we propose implementing our transformer architecture that incorporates spatial and temporal transformer encoders, along with a joint group transformer encoder and fusion module, to facilitate the lifting process in a formal context effectively.

As shown in Figure 3.1a, the input data is initially processed through an embedding module (EM), followed by pose temporal transformer encoder (PTTE), joint group transformer encoder (JGTE), and joint spatial transformer encode (JSTE). As shown in Figure 3.1b, the PTTE, JGTE and JSTE process different parts of data. The PTTE gets the entire pose sequence as input. The JGTE partitions the pose into three joint groups and processes three different joint groups' sequences. The JSTE just gets one pose as input. Subsequently, the outputs of the three transformer encoders are passed through a fusion module (FM), which fuses the output of the previous three types of transformers. The output of the FM will respectively go through regression head (RH) and center frame extraction module (CFEM), therefore finally getting the predicted 3D pose sequence and predicted center frame of the 3D pose sequence.

3.0.1 Embedding Module

The token embedding module (EM) employs a trainable convolutional layer to project each token into a higher-dimensional feature space. Each 2D pose is treated as an input token for the embedding module, analogous to the approach used in ViT. The input sequence for the embedding has a dimensionality of $X \in R^{f\times(J\times 2)}$, f is the number of frames of the input sequence, J is the number of joints of each 2d poses, and 2 indicates joint's coordinate in 2D space. While the output sequence from the embedding possesses a dimensionality of $X_e \in R^{f\times(J\times C)}$, C is the embedding output channel. The process can be expressed as follows:

$$X_e = CONV(X) = CONV([x_1, x_2; ...; x_f]) \tag{3.1}$$

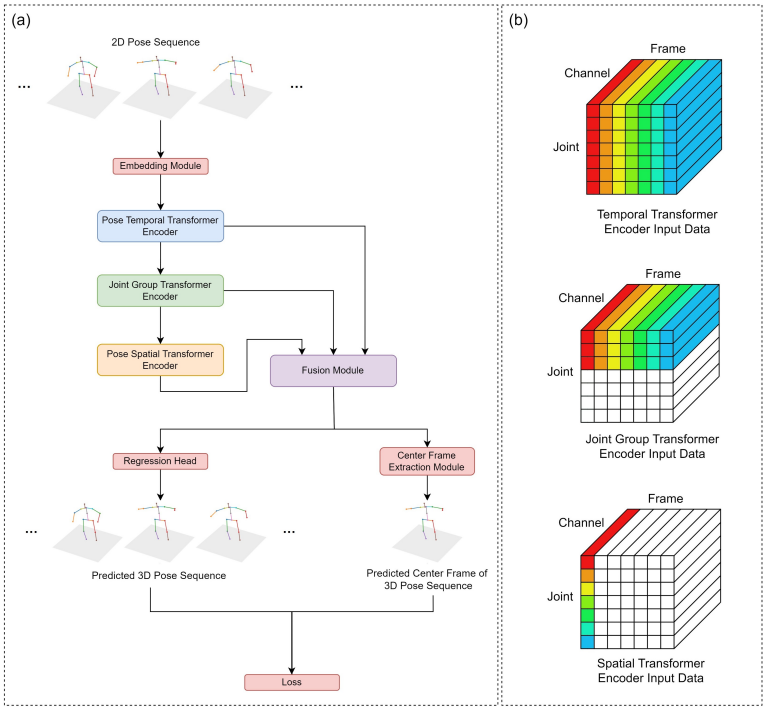

Figure 3.1: (a) The entire 3d pose estimation process. The PTTE predicts the pose temporal correlation between each frame. The JSTE predicts the spatial correlation between each skeleton joint in one frame. The JGTE partitions the input human skeleton into three parts (joint groups) based on motion coordination pose grammar and predicts the temporal and spatial correlation between each joint group. (b) Input data for PTTE, JGTE and JSTE. About the input data shape, it is $R^{f\times(J\times C)}$ for PTTE; $R^{f\times(G\times C)}$ for JGTE; $R^{J\times C}$ for JSTE. Where f is the frame number in one pose sequence, J is the joint number in one human, G is the joint number in one joint group, C is the feature dimension.

where $x_i \in R^{1\times(J\times 2)}|(i = 1, 2, ...f)$ indicates the input vector of each frame and x_i contains the information of one 2d pose in one frame. $CONV$ means the convolutional neural network computation.

3.0.2 Pose Temporal Transformer Encoder

To get the temporal pose correlation between each frame, we adopt a pose temporal transformer encoder (PTTE) like the temporal transformer encoder in [47]. As shown in Figure 3.2a, the PTTE uses positional embedding to maintain the positional information of the sequence. The architecture of PTTE is shown in Figure 3.2c. The process of the embedding module (EM) and summing with positional

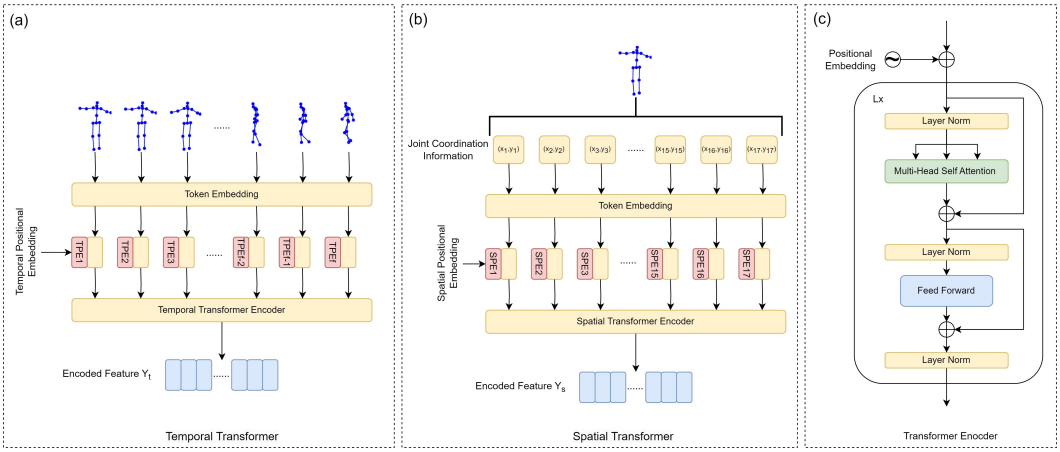

Figure 3.2: (a) The pose temporal transformer encoder (PTTE) architecture. It is inspired by the work[47]. The PTTE takes the 3D pose sequence as a token ($R^{f\times(J\times C)}$) and output encoded feature $Y_t^{f\times(J\times C)}$. It extracts features by considering pose correlations in the pose sequence. Y_t contains information of a sequence pose. (b) Spatial transformer baseline. In this thesis, the JSTE takes the 3D pose of one frame as a token ($R^{J\times C}$) and output encoded feature $Y_s^{J\times C}$. It obtains the correlation of joint correlations of each individual skeleton. Y_s contains information of a sequence pose. (c) the transformer encoder architecture for PTTE and JSTE.

embedding can be expressed as follows:

$$X_t = X_e + E_{tpos} \tag{3.2}$$

where $E_{tpos} \in R^{f\times(J\times C)}$ is the positional embedding. After going through the EM and summing with the positional embedding, we get the PTTE input X_t. The input sequence $X \in R^{f\times(J\times C)}$ becomes $X_t \in R^{f\times(J\times C)}$, where C is the embedding dimension. X_t is sent to the layer normalization module and multi-head self-attention.

Scaled Dot-Product Attention is employed to compute attention weights for each position in the input sequence. As shown in Figure 3.3b, it functions as a mapping function that maps a query matrix Q, key matrix K, and value matrix V to an output attention matrix. Q, K and V have dimensions of $N \times d$, where N denotes the number of vectors in the sequence, and d represents the dimension. A scaling factor of $\frac{1}{d}$ is utilized within this attention operation for appropriate normalization, preventing extremely small gradients when large values of d lead dot products to grow large in magnitude. Thus the output of the scaled dot-product attention can be expressed as:

$$Attention(Q, K, V) = Softmax(QK^T/\sqrt[2]{d})V \tag{3.3}$$

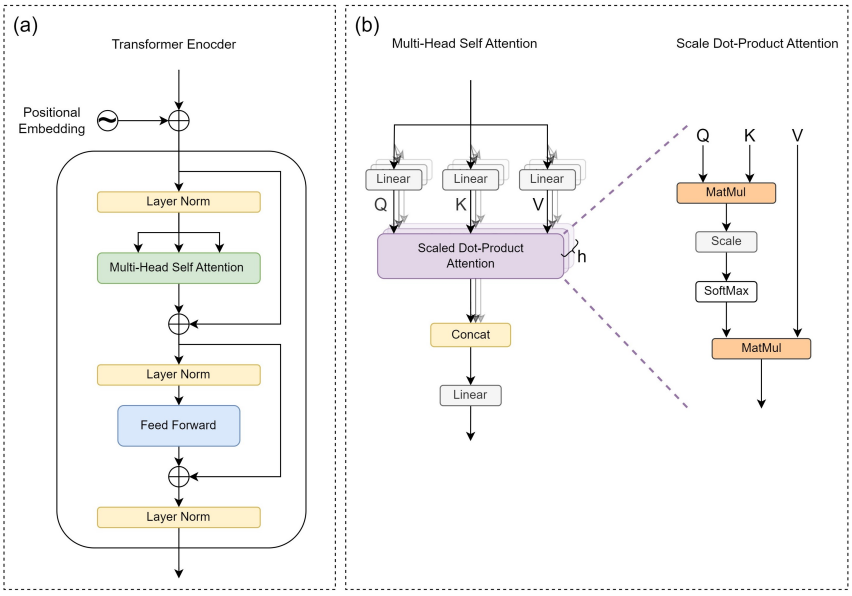

Figure 3.3: (a)The architecture of transformer encoder. The illustration of the transformer encoder is followed by ViT[8] (b)The architecture of multi-head self-attention(MSA) and scale dot-product attention. They are all used in PTTE and JSTE. h is the head number in MSA.

In the PTTE, $d = (J \times C)/h$, where J is joint number, C is the embedding dimension and h is head number of multi-head self-attention. $N = f$. The Q, K, V are computed from the embedded feature $Z \in R^{f \times C}$ by linear transformations W_Q, W_K and $W_V \in R^{C \times C}$:

$$Q = XW_Q, \quad K = XW_K, \quad QV = XW_V \tag{3.4}$$

Multi-head Self Attention Layer (MSA) is similar to vanilla transformer encoder attention in [20]. It utilizes multiple heads to model the information jointly from various representation subspaces with different positions. Each head applies scaled dot-product attention in parallel. The MSA partitions the input data $X_{t,in}$ into h heads, allowing each head to independently process the attention mechanism in parallel, as shown in Figure 3.3b.

$$MSA(Q, K, V) = Concat(H_1, ..., H_h)W_{out} \tag{3.5}$$

$$where \quad H_i = Attention(Q_i, K_i, V_i), \quad i \in [1...h] \tag{3.6}$$

Subsequently, the independent attention outputs are concatenated and linearly transformed into the expected dimension. The concatenated output is then processed through the feed-forward network (FFN). For the PTTE structure in l-th layer, the data process can be expressed as follows:

$$X'_{t,l} = MSA(LN(X_{t,l-1})) + X_{t,l-1}, l = 1, 2, ...L \tag{3.7}$$

$$X''_{t,l} = FFN(LN(X'_{t,l})) + X'_{t,l}, l = 1, 2, ...L \tag{3.8}$$

$$Y_t = LN(X''_{t,l}) \tag{3.9}$$

where $X_{t,l-1}$ is the $l-1$-th layer output or l-th layer input, $LN()$ denotes the layer normalization operator. FFN module normally contains 2 fully connected layers, 2 activation layers and 1 dropout layer. The PTTE output is $Y_t \in R^{f\times(J\times C)}$. The PTTE consists of L identical layers and the PTTE output $Y_t \in R^{f\times(J\times C)}$ keeps the same size as PTTE input $X_t \in R^{f\times(J\times C)}$.

3.0.3 Joint Group Transformer Encoder

Inspired by [9] and [21], a joint group transformer encoder (JGTE) receives Y_t and captures the spatial-temporal correlations between joint groups. The process is shown in Figure 3.4. The JGTE just has one encoder layer.

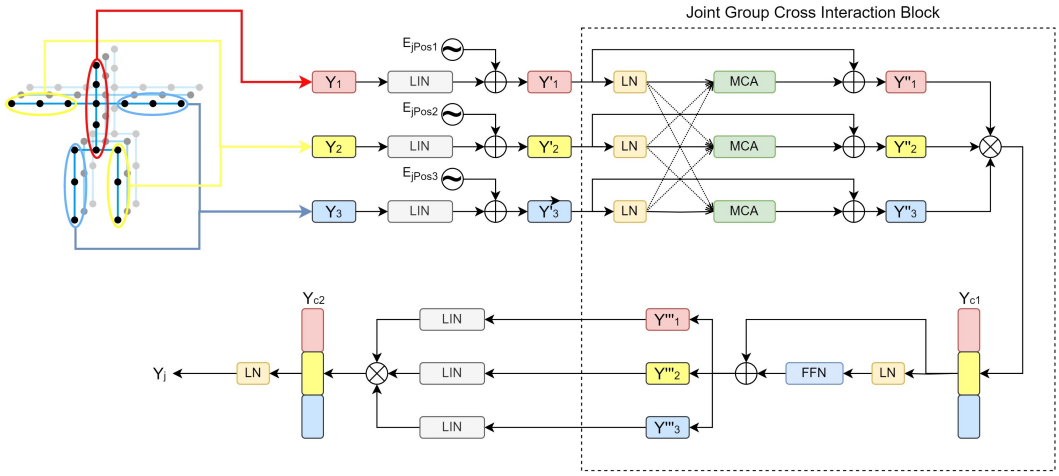

Figure 3.4: Joint Group Transformer Encoder (JGTE) architecture. It is designed to measure both spatial and temporal correlations between joint groups. LIN is a linear layer, and LN is a layer normalization layer, MCA is multi-head cross-attention, FFN is feed-forward network.

Before implementing the JGTE, the PTTE output($Y_t \in R^{f\times(J\times C)}$) is partitioned into 3 parts($Y_1 \in R^{f\times(5\times C)}, Y_2 \in R^{f\times(6\times C)}, Y_3 \in R^{f\times(6\times C)}$) following the motor coordination grammar[9] shown in Figure 2.1.

$$Y_1, Y_2, Y_3 = DIV(Y_t) \tag{3.10}$$

Among them, Y_1 encapsulates the human torso temporal-spatial information; Y_2 contains human left leg and right arm temporal-spatial information; Y_3 contains human right leg and left arm temporal-spatial information.

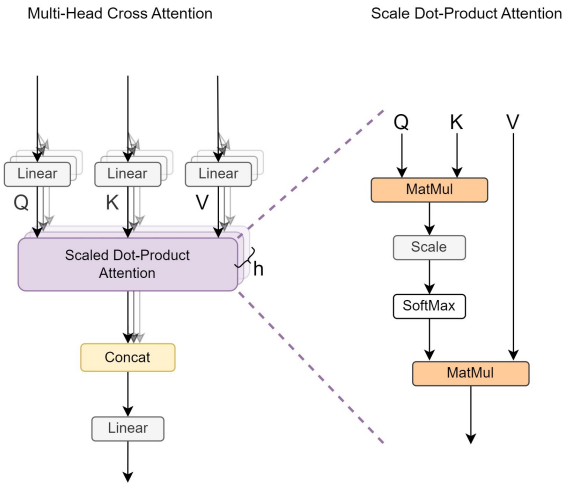

Figure 3.5: Multi-Head Cross-Attention (MCA).

Then to record the temporal and relative positional information of each joint in each group, those three data will go through linear projection and sum with three corresponding positional embeddings $E_{jPos1} \in R^{f\times(5\times C)}$, $E_{jPos2} \in R^{f\times(6\times C)}$ and $E_{jPos3} \in R^{f\times(6\times C)}$ respectively:

$$Y_1^{'} = LIN(Y_1) + E_{jPos1} \tag{3.11}$$

$$Y_2^{'} = LIN(Y_2) + E_{jPos2} \tag{3.12}$$

$$Y_3^{'} = LIN(Y_3) + E_{jPos3} \tag{3.13}$$

After the initial processing, the summed results are subjected to layer normalization and multi-head cross-attention (MCA) mechanisms, which measure the temporal-spatial correlation among joint groups and has a similar structure to MSA, as shown in Figure 3.5. The MSA focuses on a specific part of input data, but MCA is used to get the relation of different input data. For each cross-attention the input data $Y_1^{'}$, $Y_2^{'}$ and $Y_3^{'}$ are alternatively regarded as query (Q), key (k) and value (V).

$$Y_1^{''} = Y_1^{'} + MCA(LN(Y_1^{'}), LN(Y_2^{'}), LN(Y_3^{'})) \tag{3.14}$$

$$Y_2^{''} = Y_2^{'} + MCA(LN(Y_2^{'}), LN(Y_1^{'}), LN(Y_3^{'})) \tag{3.15}$$

$$Y_3^{''} = Y_3^{'} + MCA(LN(Y_3^{'}), LN(Y_2^{'}), LN(Y_1^{'})) \tag{3.16}$$

After concatenating $Y_1^{''}$, $Y_2^{''}$ and $Y_3^{''}$ we get Y_{c1}.

$$Y_{c1} = Concate(Y_1^{''}, Y_2^{''}, Y_3^{''}) \tag{3.17}$$

Before getting the final output Y_j of joint group transformer encoder, Y_{c1} must be processed by feed-forward network (FFN) and be divided into $Y_1^{'''}$, $Y_2^{'''}$ and $Y_3^{'''}$. $Y_1^{'''}$, $Y_2^{'''}$ and $Y_3^{'''}$ will be mapped to their corresponding trainable linear projection and get concatenated into Y_{c2} which will be Y_j after going through a layer normalization. The process can be expressed as follows:

$$Y_1^{'''}, Y_2^{'''}, Y_3^{'''} = DIV(Y_{c1} + FFN(LN(Y_{c1}))) \tag{3.18}$$

$$Y_j = LN(Concate(LIN(Y_1^{'''}), LIN(Y_2^{'''}), LIN(Y_3^{'''}))) \tag{3.19}$$

3.0.4 Joint Spatial Transformer Encoder

Joint spatial transformer encoder (JSTE) is adopted to get the joint spatial correlation in one frame. The architecture of the JSTE is like the temporal transformer encoder except for the input data format. In the PTTE, the input data is $X_t \in R^{f\times(J\times C)}$, but in the JSTE, the input data is $X_s \in R^{J\times C}$, where J is the joint number and C is the feature embedding dimension, as shown in Figure 3.2b. The architecture of JSTE is shown in Figure 3.2c. The JSTE takes each joint information as a token, but the PTTE takes one pose information as a token. Following the general transformer pipeline to perform the feature extraction among all tokens. Those tokens are sent to the JSTE after summing with spatial positional embedding. And finally, get the JSTE result Y_s. The JSTE consists of L identical layers and the JSTE output $Y_s \in R^{J\times C}$ keeps the same size as JSTE input $X_s \in R^{J\times C}$.

3.0.5 Fusion Module

Inspired by [21], the fusion module (FM) is used to fuse the output of PTTE $Y_t \in R^{f\times(J\times C)}$, JGTE $Y_j \in R^{f\times(J\times C)}$ and JSTE $Y_s \in R^{f\times(J\times C)}$. The architecture of FM is the same as the architecture of the JGTE except for the input difference. In the JGTE, we regard each joint group (Y_1, Y_2 and Y_3) as input, but in the FM, the inputs are Y_t, Y_s and Y_j which are the outputs of previous three types of transformer encoder. The output of FM encoder is denoted as $Y_f \in R^{f\times(J\times C\times 3)}$

3.0.6 Regression Head for 3D Pose Sequence

The regression Head (RH) is used to predict 3d coordinates of the pose sequence. In the RH, three linear transformation layers are applied on the FM output Y_f to perform regression to produce the 3D pose sequence $Y_{seq} \in R^{f\times J\times 3}$, where f is the frame number, J is the joint number, 3 is the joint 3d coordinate. Finally, the output 3D pose sequence Y_{seq} is used to compute the loss.

3.0.7 Center Frame Extraction Module

The center frame extraction module (CFEM) works as a full-to-singe scheme like strided transformer[20] does. The CFEM contains 4 convolution layers, where 3 are used to reduce the frame length and 1 is used as a RH for a single 3d pose. This module further refines the output from the FM to produce more accurate estimations. The output of CFEM is denoted as $Y_{center} \in R^{1\times J\times 3}$, where 1 is the center frame of pose sequence, J is the joint number, 3 is the joint 3d coordinate.

3.0.8 Loss Function

Our model has two outputs which are 3D pose sequence and 3D pose center frame. We compute mean Squared Error (MSE) loss for 3D pose sequence and 3D pose center frame. The MSE loss for the 3D pose sequence can be defined as:

$$L_{seq} = \sum_{i=1}^{T}\sum_{J=1}^{J} \|Y^i_{seq,j} - \widetilde{Y^i_{seq,j}}\|_2 \tag{3.20}$$

where the $Y^i_{seq,j}$ is the estimated 3d coordination of j-th joint in i-th frame and $\widetilde{Y^i_{seq,j}}$ is the corresponding ground truth. The MSE loss for the center frame of 3D pose sequence can be defined as:

$$L_{center} = \sum_{i=1}^{T}\sum_{J=1}^{J} \|Y^i_{center,j} - \widetilde{Y^i_{center,j}}\|_2 \tag{3.21}$$

where the $Y^i_{center,j}$ is the estimated 3d coordination of j-th joint in i-th frame and $\widetilde{Y^i_{center,j}}$ is the corresponding ground truth.

Given two loss outputs, L_{seq} and L_{center}, and their corresponding weights W_{seq} and W_{center}, the final loss L can be computed as a weighted sum of the individual losses:

$$L = L_{seq} \times W_{seq} + L_{center} \times W_{center} \tag{3.22}$$

In our experiment W_{seq} and W_{center} are both 0.5.

Chapter 4

Experiments

4.0.1 Datasets and Evaluation Metrics

We evaluate our model on two commonly used 3DHPE datasets: Human3.6M[15] and MPI-INF-3DHP[28].

Human3.6M. It is the most widely used indoor dataset for 3D single person HPE. The dataset includes 11 professional actors performing 17 different actions, such as sitting, walking, and talking on the phone. The videos of each subject were recorded from 4 different views in an indoor environment. This dataset contains 3.6 million video frames with 3D ground truth annotation captured by an accurate marker-based motion capture system. This makes it an ideal dataset for training and evaluating HPE algorithms. Following the same policy of others [[31], [11], [25], [21]], 5 subjects (S1, S5, S6, S7, S8) are used for training, and 2 subjects (S9, S11) are used for testing.

MPI-INF-3DHP. The MPI-INF-3DHP dataset, which contains 1.3 million frames and features a wider range of motions than Human3.6M, is also extensively used for 3D human pose estimation due to its large scale and the increased challenge posed by its inclusion of both indoor and intricate outdoor settings. The training set consists of 8 subjects performing 8 different activities, while the test set comprises 7 subjects. The same as SOTAs [[21], [32], [43], [47]], we train our method using the training set and evaluate it using the valid frames in the test set.

Evaluation Metrics. Following [[21], [32], [43], [47]], we use the same metrics for performance evaluation. For Human3.6M, two evaluation protocols are adopted to calculate the quantitative results. Protocol #1 measures the mean Euclidean distance in millimeters (mm) between the predicted 3D poses and the ground truth 3D poses and is referred to as Mean Per Joint Position Error (MPJPE). Protocol #2 refers to P-MPJPE which is the MPJPE between aligned 3D pose predictions and ground truths. For MPI-INF-3DHP, evaluation metrics such as the area under the curve (AUC), percentage of correct keypoints (PCK), and mean per-joint position error (MPJPE) are employed.

4.0.2 Implementation Details

We implement our proposed method with Pytorch. For Human3.6M, we trained our JSTFormer from scratch for 200 epochs on 5 NVIDIA RTX 3090 GPUs using AdamW optimizer with an initial learning rate of 0.0016 and learning rate decay of 0.99. The batch size was 800 for each GPU. We chose frame sequence length to be 81 and applied pose flipping horizontally as data augmentation both in training and testing following [30].

Because, in Human3.6M, the poses from adjacent frames usually contain redundant information since few changes happen between them if the video FPS is large (25 or 30). Hence, to increase the diversity of the input poses, one way is to collect a longer pose sequence. But this will increase the computation burden. Following [31], we collected a pose sequence with a fixed number of frames T=81 and sampled the frames with an interval of 5 in order to cover more temporal information with fixed computation cost.

For the 2D pose detector, we used the cascaded pyramid network (CPN)[8] on Human3.6M following [30], and also used the ground truth 2D pose as input for Human3.6M.

For MPI-INF-3DHP, we trained our JSTFormer from scratch for 50 epochs on 5 NVIDIA RTX 3090 GPUs using AdamW optimizer with an initial learning rate of 0.0016 and learning rate decay of 0.98. But we didn't sample the frames with an interval.

4.0.3 Results and Comparisons

Human3.6M. We evaluate our method under two protocols. Tables 4.1 and 4.2 present the outcomes for Protocol 1 of our JSTFormer model, which is trained on CPN predictions and GT, respectively. By employing both CPN and GT predictions as input, our model achieves competitive performance.

We compare our method with 10 state-of-the-art methods and report quantitative comparisons of protocol #1 with input 2d pose detected by the cascaded pyramid network (CPN) in Table 4.1. It can be seen that our method achieves the 2nd best performance. Additionally, our average MPJPE (45.6mm) outperforms the 3th best result (46.8mm) by 1.2%. Our average MPJPE (45.6mm) is worse than the best result (43.0mm) by 2.6%. In Eating, Phoneing, Photoing, Siting, Smoking, Waiting, Walking Down, and Walking actions, our method reaches the second-best performence.

In Table 4.2, we compare our results with 10 state-of-the-art methods and report quantitative comparisons of protocol #1 with ground truth (GT) input 2d pose. It can be seen that our method achieves the 3rd best performance and our average MPJPE (33.8mm) is worse than the best result (30.5mm) by 3.2%.

In Table 4.3, we compare our results with 9 state-of-the-art methods and report quantitative comparisons of protocol #1 with ground truth (GT) input 2d pose and

Table 4.1: Protocol 1 with MPJPE (mm): Reconstruction error on Human3.6M. Input 2D joints are acquired by detection. CPN - Cascaded Pyramid Network. Red: best; Blue: second best. T is the input length. For MPJPE : the lower the better.

Method(CPN)		Dir	Disc	Eat	Greet	Phone	Photo	Pose	Purch	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Average
Martinez *et al.*[27]	ICCV'17	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Hossain *et al.*[14]	ECCV'18	48.4	50.7	57.2	55.2	63.1	72.6	53.0	51.7	66.1	80.9	59.0	57.3	62.4	46.6	49.6	58.3
Zhao *et al.*[45]	CVPR'19	47.3	60.7	51.4	60.5	61.1	49.9	47.3	68.1	86.2	55.0	67.8	61.0	60.6	42.1	45.3	57.6
Luvizon *et al.*[26]	CVPR'18	49.2	51.6	47.6	50.5	51.8	60.3	48.5	51.7	61.5	70.9	53.7	48.9	57.9	44.4	48.9	53.2
Lee *et al.*[18]	ECCV'18	40.2	49.2	47.8	52.6	50.1	75.0	50.2	43.0	55.8	73.9	54.1	55.6	58.2	43.3	43.3	52.8
Dabral *et al.*[6]	ECCV'18	44.8	50.4	44.7	49.0	52.9	61.4	43.5	45.5	63.1	87.3	51.7	48.5	52.2	37.6	41.9	52.1
GraphSH[40](T=1)	CVPR'21	45.2	49.9	47.5	50.9	54.9	66.1	48.5	46.3	59.7	71.5	51.4	48.6	53.9	39.9	44.1	51.9
GraFormer[46](T=1)	CVPR'22	45.2	50.8	48.0	50.0	54.9	65.0	48.2	47.1	60.2	70.0	51.6	48.7	54.1	39.7	43.1	51.8
MGCN[48](T=1)	ICCV'21	45.4	49.2	45.7	49.4	50.4	58.2	47.9	46.0	57.5	63.0	49.7	46.6	52.2	38.9	40.8	49.4
Pavllo *et al.*[30]	CVPR'19	45.2	46.7	43.3	45.6	48.1	55.1	44.6	44.3	57.3	65.8	47.1	44.0	49.0	32.8	33.9	46.8
UGCN[37](T=96)	ECCV'20	41.3	43.9	44.0	42.2	48.0	57.1	42.2	43.2	57.3	61.3	47.0	43.5	47.0	32.6	31.8	45.6
MHFormer[21](T=351)	CVPR'22	39.2	43.1	40.1	40.9	44.9	51.2	40.6	41.3	53.5	60.3	43.7	41.1	43.8	29.8	30.6	43.0
JSTFormer(T=81)	Ours	42.3	45.3	43.0	44.5	47.3	54.4	43.6	44.1	55.7	63.1	46.6	42.9	46.3	31.8	32.7	45.6

Table 4.2: Protocol 1 with MPJPE (mm): Reconstruction error on Human3.6M. Input 2D joints are ground truth 2D poses. Red: best; Blue: second best. T is the input length. For MPJPE : the lower the better.

Method(GT)		Dir	Disc	Eat	Greet	Phone	Photo	Pose	Purch	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Average
Wandt *et al.*[36]	CVPR'19	50.0	53.5	44.7	51.6	49.0	58.7	48.8	51.3	51.1	66.0	46.6	50.6	42.5	38.8	60.4	50.9
Martinez *et al.*[27]	ICCV'17	37.7	44.4	40.3	42.1	48.2	54.9	44.4	42.1	54.6	58.0	45.1	46.4	47.6	36.4	40.4	45.5
Zhao *et al.*[45]	CVPR'19	37.8	49.4	37.6	40.9	45.1	41.4	40.1	48.3	50.1	42.2	53.5	44.3	40.5	47.3	39.0	43.8
Hossain *et al.*[14]	ECCV'18	35.2	40.8	37.2	37.4	43.2	44.0	38.9	35.6	42.3	44.6	39.7	39.7	40.2	32.8	35.5	39.2
VPose[30](T=243)	CVPR'19	35.2	40.2	32.7	35.7	38.2	45.5	40.6	36.1	48.8	47.3	37.8	39.7	38.7	27.8	29.5	37.8
GraFormer[46](T=1)	CVPR'22	32.0	38.0	30.4	34.4	34.7	43.3	35.2	31.4	38.0	46.2	34.2	35.7	36.1	27.4	30.6	35.2
Liu *et al.*[24](T=243)	CVPR'20	34.5	37.1	33.6	34.2	32.9	37.1	39.6	35.8	40.7	41.4	33.0	33.8	33.0	26.6	26.9	34.7
Ray3D *et al.*[42](T=9)	CVPR'22	31.2	35.7	31.4	33.6	35.0	37.5	37.2	30.9	42.5	41.3	34.6	36.5	32.0	27.7	28.9	34.4
PoseFormer[47](T=81)	ICCV'21	30.0	33.6	29.9	31.0	30.2	33.3	34.8	31.4	37.8	38.6	31.7	31.5	29.0	23.3	23.1	31.3
MHFormer[21](T=351)	CVPR'22	27.7	32.1	29.1	28.9	30.0	33.9	33.0	31.2	37.0	39.3	30.0	31.0	29.4	22.2	23.0	30.5
JSTFormer(T=81)	Ours	31.1	34.1	33.7	31.8	34.4	37.4	34.4	34.2	41.0	43.8	34.3	32.7	33.4	25.4	25.4	33.8

Table 4.3: Protocol 2 with P-MPJPE (mm): Reconstruction error on Human3.6M with similarity transformation. CPN - Cascaded Pyramid Network. GT - Ground truth. Red: best; Blue: second best. T is the input length. For P-MPJPE : the lower the better.

Method(CPN)		Dir	Disc	Eat	Greet	Phone	Photo	Pose	Purch	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Average
Martinez *et al.*[27]	ICCV'17	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	69.4	49.2	45.0	49.5	38.0	43.1	47.7
Hossain *et al.*[14](GT)	ECCV'18	35.7	39.3	44.6	43.0	47.2	54.0	38.3	37.5	51.6	61.3	46.5	41.4	47.3	34.2	39.4	44.1
ST-GCN[2](T=7)	ICCV'19	35.7	37.8	36.9	40.7	39.6	45.2	37.4	34.5	46.9	50.1	40.5	36.1	41.0	29.6	33.2	39.0
SGNN[41](T=9)	ICCV'21	33.9	37.2	36.8	38.1	38.7	43.5	37.8	35.0	47.2	53.8	40.7	38.3	41.8	30.1	31.4	39.0
Cai *et al.*[2](T=7)(GT)	ICCV'19	35.7	37.8	36.9	40.7	39.6	45.2	37.4	34.5	46.9	50.1	40.5	36.1	41.0	29.6	32.3	39.0
Wandt *et al.*[36](GT)	CVPR'19	33.6	38.8	32.6	37.5	36.0	44.1	37.8	34.9	39.2	52.0	37.5	39.8	34.1	40.3	34.9	38.2
Lin *et al.*[22](GT)	BMVC'19	32.5	35.3	34.3	36.2	37.8	43.0	33.0	32.2	45.7	51.8	38.4	32.8	37.5	25.8	28.9	36.8
Pavllo *et al.*[30](T=243)(CPN)	CVPR'19	34.1	36.1	34.4	37.2	36.4	42.2	34.4	33.6	45.0	52.5	37.4	33.8	37.8	25.6	27.3	36.5
P-STMO[32](T=243)(CPN)	ECCV'22	31.3	35.2	32.9	33.9	35.4	39.3	32.5	31.5	44.6	48.2	36.3	32.9	34.4	23.8	23.9	34.4
JSTFormer(T=81)(CPN)	Ours	35.2	35.5	36.1	36.1	36.1	36.8	36.7	39.5	44.7	44.4	40.9	34.1	30.9	32.3	30.4	36.6

detected input 2d pose by the cascaded pyramid network (CPN). Although our method reaches the 3 best performance (36.6mm), the gap between our method and the 2nd best result (36.5mm) is just 0.1mm. In some action like Photoing, Siting Down and Walking Down, out method has the best performance.

MPI-INF-3DHP. We compare our method with 7 state-of-the-art methods ([22], [5], [37], [21], [43], [32]). The quantitative comparisons on MPI-INF-3DHP are reported in Table 4.4. As seen, although there is a gap between the best result, our results achieve the 2nd best performance and outperform previous other methods across the vast majority of subjects and on average.

Table 4.4: Quantitative comparison with the state-of-the-art methods on MPI-INF-3DHP under three metrics. ↑ indicates the higher, the better. ↓ indicates the lower, the better. Red: best; Blue: second best. T is the input length.

Method		PCK ↑	AUC ↑	MPJPE ↓
Lin *et al.*[22](T=25)	BMVC'19	83.6	51.4	79.8
Chen *et al.*[5](T=243)	TCSVT'21	87.8	53.8	79.1
PoseFormer[47](T=9)	ICCV'21	88.6	56.4	77.1
Wang *et al.*[37](T=96)	ECCV'20	86.9	62.1	68.1
MHFormer[21](T=9)	CVPR'22	93.8	63.3	58.0
MixSTE[43](T=27)	CVPR'22	94.4	66.5	54.9
P-STMO[32](T=81)	ECCV'22	97.9	75.8	32.2
JSTFormer(T=81)	Ours	97.6	72.2	37.6

4.0.4 Ablation Study

Extensive ablation experiments have been performed on Human3.6M dataset using CPN poses as input and MPJPE (mm) as the evaluation metric to examine the impact of various modules and hyperparameters.

Effect of Architecture Modules. We first study the module choices by configuring our modules with different combinations. The results are reported in Table 4.5. It can be seen that the neural network get the best result when JGTE, Fusion and Refinement are used. The JGTE plays an important role in the model. The model gets the largest improvement when adding the Fusion module.

Table 4.5: Ablation study on different encoder combinations. PTTE : Pose Temporal Transformer Encoder, JSTE : Joint Spatial Transformer Encoder, JGTE : Joint Group Transformer Encoder

Model	PTTE	JSTE	JGTE	Fusion	Refinement	MPJPE
1	✓	✓	✗	✗	✗	48.1
2	✓	✓	✓	✓	✗	48.0
3	✓	✓	✗	✓	✓	47.7
4	✓	✓	✓	✗	✓	47.1
5	✓	✓	✓	✓	✓	46.8

Effect of Architecture Hyper-Parameters. Table 4.6 reports the results of different settings of the hyper-parameter T (the input length) and N (the sample interval). It is obvious that when T = 81, N = 5, it gets the best result. Bigger interval tends to get a better result.

Table 4.6: Ablation study for hyper-parameter setting in the input length (T) and interval (N).

T	81			243			351		
N	1	3	5	1	3	5	1	3	5
MPJPE	48.2	46.5	45.6	46.8	48.1	48.1	51.0	47.7	49.2

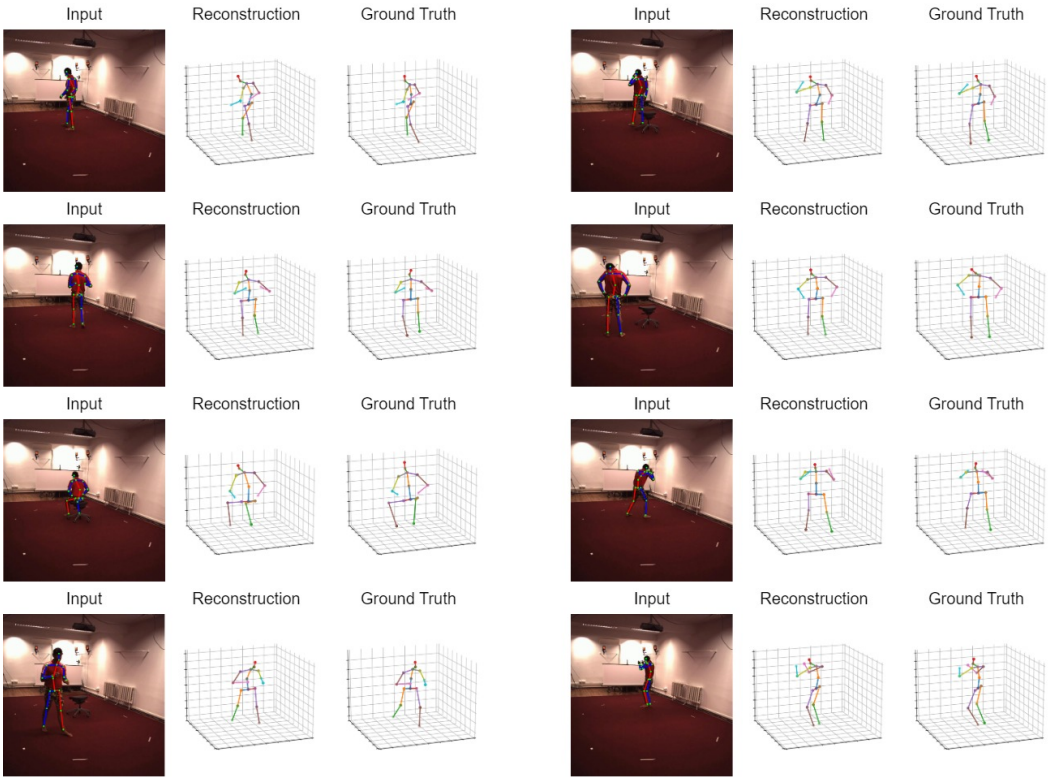

Figure 4.1: Qualitative results for several actions in the Humans3.6M dataset. From left to right: Original RGB image with 2D keypoint predictions using hrnet. 3D reconstruction using our method (T=81) which uses hrnet output 2d keypoints as input data. Ground truth 3D keypoints.

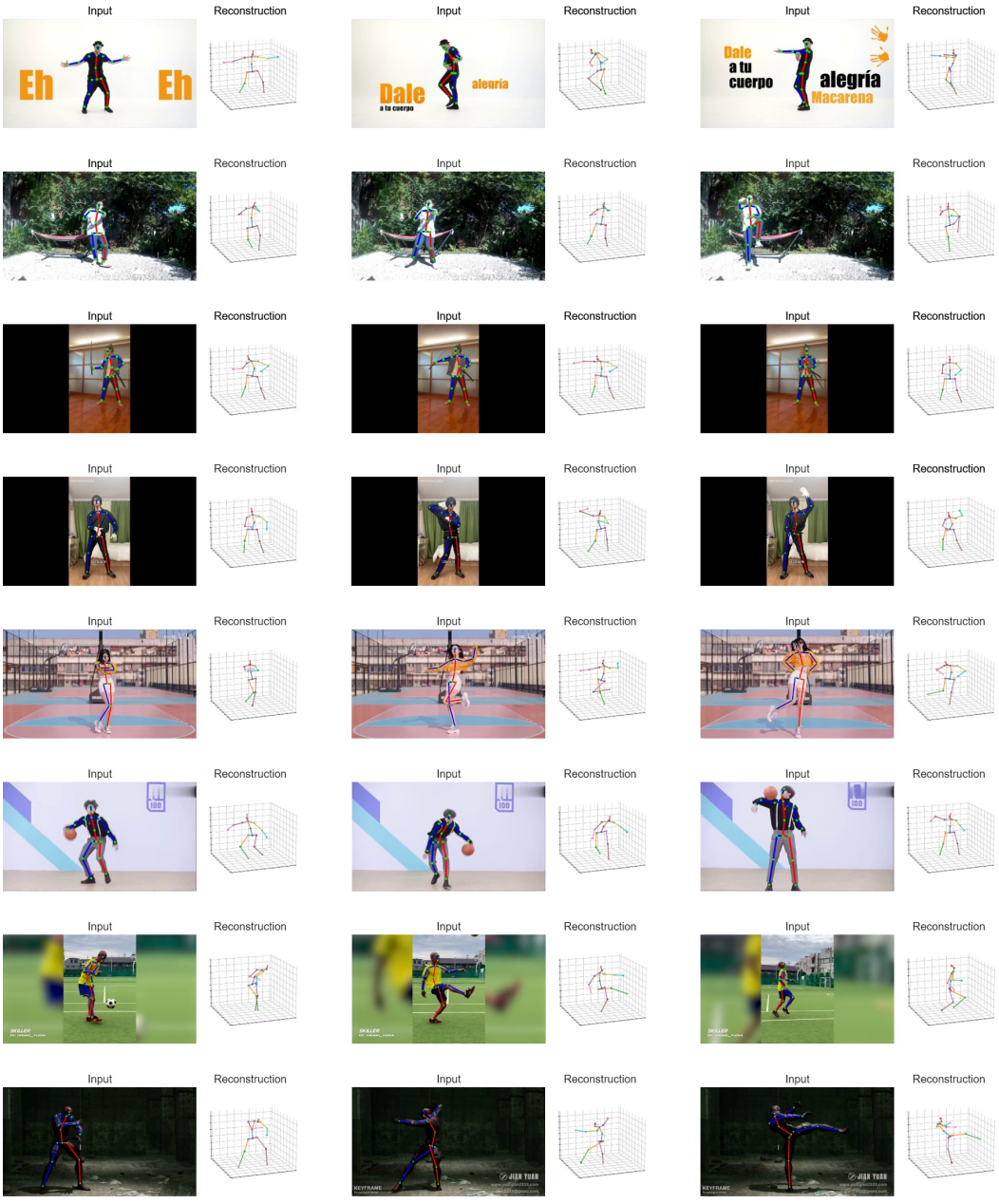

Figure 4.2: Qualitative results for in the wild scenes.

Chapter 5

Conclusion

In this thesis, we present JSTFormer, a transformer that computes the temporal and spatial correlation between each joint group by taking the joint groups as query, key, and values into the attention respectively. We also combine the excellent work of [31], [21] and [47]. The pose temporal transformer encoder captures the temporal relationships between the poses in each frame, while the joint spatial transformer focuses on the spatial correlations among each joint. Incorporating the concept of residual blocks, we develop a fusion module that combines the outputs of the temporal, spatial, and joint group transformer encoder to refine the ultimate pose sequences. To refine the fusion module output, we also use a center frame extraction module to predict the center frame of the pose sequence. Our method demonstrates competitive performance on Human3.6m and MPI-INF-3DHP datasets.

There are some limitations in our work. When the human pose is unusual (like people upside down, people curl up), the prediction fails. This work cannot predict the video in real-time. For the 10s 60FPS video, it will take about 1 minute to predict it. When there is camera obstruction, the network cannot get precise results on the human-occluded parts.

Bibliography

[1] Aritz Badiola-Bengoa and Amaia Mendez-Zorrilla. "A systematic review of the application of camera-based human pose estimation in the field of sport and physical exercise". In: *Sensors* 21.18 (2021), p. 5996.

[2] Yujun Cai et al. "Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 2272–2281.

[3] Ching-Hang Chen and Deva Ramanan. "3d human pose estimation= 2d pose estimation+ matching". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7035–7043.

[4] Lili Chen et al. "Decision transformer: Reinforcement learning via sequence modeling". In: *Advances in neural information processing systems* 34 (2021), pp. 15084–15097.

[5] Tianlang Chen et al. "Anatomy-aware 3d human pose estimation with bone-based pose decomposition". In: *IEEE Transactions on Circuits and Systems for Video Technology* 32.1 (2021), pp. 198–209.

[6] Rishabh Dabral et al. "Learning 3d human pose from structure and motion". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 668–683.

[7] Linhao Dong, Shuang Xu, and Bo Xu. "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition". In: *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2018, pp. 5884–5888.

[8] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[9] Hao-Shu Fang et al. "Learning pose grammar to encode human body configuration for 3d pose estimation". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.

[10] Rohit Girdhar et al. "Video action transformer network". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 244–253.

[11] Kehong Gong, Jianfeng Zhang, and Jiashi Feng. "Poseaug: A differentiable pose augmentation framework for 3d human pose estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8575–8584.

[12] Kai Han et al. "A survey on vision transformer". In: *IEEE transactions on pattern analysis and machine intelligence* 45.1 (2022), pp. 87–110.

[13] Michael Hofmann and Dariu M Gavrila. "Multi-view 3D human pose estimation in complex environment". In: *International journal of computer vision* 96 (2012), pp. 103–124.

[14] Mir Rayat Imtiaz Hossain and James J Little. "Exploiting temporal information for 3d human pose estimation". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 68–84.

[15] Catalin Ionescu et al. "Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments". In: *IEEE transactions on pattern analysis and machine intelligence* 36.7 (2013), pp. 1325–1339.

[16] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. "Visualizing and understanding recurrent networks". In: *arXiv preprint arXiv:1506.02078* (2015).

[17] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. "VIBE: Video Inference for Human Body Pose and Shape Estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[18] Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. "Propagating lstm: 3d pose estimation based on joint interdependency". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 119–135.

[19] Sijin Li and Antoni B Chan. "3d human pose estimation from monocular images with deep convolutional neural network". In: *Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part II 12*. Springer. 2015, pp. 332–347.

[20] Wenhao Li et al. "Exploiting temporal contexts with strided transformer for 3d human pose estimation". In: *IEEE Transactions on Multimedia* (2022).

[21] Wenhao Li et al. "Mhformer: Multi-hypothesis transformer for 3d human pose estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 13147–13156.

[22] Jiahao Lin and Gim Hee Lee. "Trajectory space factorization for deep video-based 3d human pose estimation". In: *arXiv preprint arXiv:1908.08289* (2019).

[23] Kevin Lin, Lijuan Wang, and Zicheng Liu. "End-to-end human pose and mesh reconstruction with transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 1954–1963.

[24] Ruixu Liu et al. "Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 5064–5073.

[25] Adrian Llopart. "Liftformer: 3d human pose estimation using attention models". In: *arXiv preprint arXiv:2009.00348* (2020).

[26] Diogo C Luvizon, David Picard, and Hedi Tabia. "2d/3d pose estimation and action recognition using multitask deep learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5137–5146.

[27] Julieta Martinez et al. "A simple yet effective baseline for 3d human pose estimation". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2640–2649.

[28] Dushyant Mehta et al. "Monocular 3d human pose estimation in the wild using improved cnn supervision". In: *2017 international conference on 3D vision (3DV)*. IEEE. 2017, pp. 506–516.

[29] Tewodros Legesse Munea et al. "The progress of human pose estimation: A survey and taxonomy of models applied in 2D human pose estimation". In: *IEEE Access* 8 (2020), pp. 133330–133348.

[30] Dario Pavllo et al. "3d human pose estimation in video with temporal convolutions and semi-supervised training". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 7753–7762.

[31] Xiaoye Qian et al. "HSTFormer: Hierarchical Spatial-Temporal Transformers for 3D Human Pose Estimation". In: *arXiv preprint arXiv:2301.07322* (2023).

[32] Wenkang Shan et al. "P-stmo: Pre-trained spatial temporal many-to-one model for 3d human pose estimation". In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*. Springer. 2022, pp. 461–478.

[33] Leonid Sigal, Alexandru O Balan, and Michael J Black. "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion". In: *International journal of computer vision* 87.1-2 (2010), p. 4.

[34] Binh Tang and David S Matteson. "Probabilistic transformer for time series analysis". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 23592–23608.

[35] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[36] Bastian Wandt and Bodo Rosenhahn. "Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 7782–7791.

[37] Jingbo Wang et al. "Motion guided 3d pose estimation from videos". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*. Springer. 2020, pp. 764–780.

[38] Qiang Wang et al. "Learning deep transformer models for machine translation". In: *arXiv preprint arXiv:1906.01787* (2019).

[39] Qingqiang Wu et al. "Human 3D pose estimation in a lying position by RGB-D images for medical diagnosis and rehabilitation". In: *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE. 2020, pp. 5802–5805.

[40] Tianhan Xu and Wataru Takano. "Graph stacked hourglass networks for 3d human pose estimation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 16105–16114.

[41] Ailing Zeng et al. "Learning skeletal graph neural networks for hard 3d pose estimation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 11436–11445.

[42] Yu Zhan et al. "Ray3D: ray-based 3D human pose estimation for monocular absolute 3D localization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 13116–13125.

[43] Jinlu Zhang et al. "Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 13232–13242.

[44] Siqi Zhang et al. "A Survey on Depth Ambiguity of 3D Human Pose Estimation". In: *Applied Sciences* 12.20 (2022), p. 10591.

[45] Long Zhao et al. "Semantic graph convolutional networks for 3d human pose regression". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 3425–3435.

[46] Weixi Zhao, Weiqiang Wang, and Yunjie Tian. "GraFormer: Graph-oriented transformer for 3D pose estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 20438–20447.

[47] Ce Zheng et al. "3d human pose estimation with spatial and temporal transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 11656–11665.

[48] Zhiming Zou and Wei Tang. "Modulated graph convolutional network for 3D human pose estimation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 11477–11487.