MixSTE: Seq2seq Mixed Spatio-Temporal Encoder for 3D Human Pose Estimation in Video

Jinlu Zhang[1] Zhigang Tu[1*] Jianyu Yang[2] Yujin Chen[3†] Junsong Yuan[4]
[1]Wuhan University [2]Soochow University [3]Technical University of Munich
[4]State University of New York at Buffalo
{jinluzhang, tuzhigang}@whu.edu.cn, jyyang@suda.edu.cn, yujin.chen@tum.de, jsyuan@buffalo.edu

Abstract

*Recent transformer-based solutions have been introduced to estimate 3D human pose from 2D keypoint sequence by considering body joints among all frames globally to learn spatio-temporal correlation. We observe that the motions of different joints differ significantly. However, the previous methods cannot efficiently model the solid inter-frame correspondence of each joint, leading to insufficient learning of spatial-temporal correlation. We propose MixSTE (Mixed Spatio-Temporal Encoder), which has a temporal transformer block to separately model the temporal motion of each joint and a spatial transformer block to learn inter-joint spatial correlation. These two blocks are utilized alternately to obtain better spatio-temporal feature encoding. In addition, the network output is extended from the central frame to entire frames of the input video, thereby improving the coherence between the input and output sequences. Extensive experiments are conducted on three benchmarks (*i.e. *Human3.6M, MPI-INF-3DHP, and HumanEva). The results show that our model outperforms the state-of-the-art approach by 10.9% P-MPJPE and 7.6% MPJPE. The code is available at* https://github.com/JinluZhang1126/MixSTE.*

1. Introduction

3D human pose estimation from monocular observations is a fundamental vision task that reconstructs 3D body joint locations from the input images or video. Since this task can obtain meaningful expressions of body geometry and motion, it has a wide range of applications, such as action recognition [54, 55], virtual human [5–7, 52], and human-robot interaction [11, 43, 50]. Most recent works are based on the 2D-to-3D lifting pipeline [1, 4, 28, 31, 37, 46, 57], which detects 2D keypoints firstly and then lift them to 3D. Due to the depth ambiguity of monocular data, multiple potential 3D poses may be mapped from the same 2D pose, so it is difficult to recover an accurate 3D pose merely based on the information of a single frame 2D keypoints.

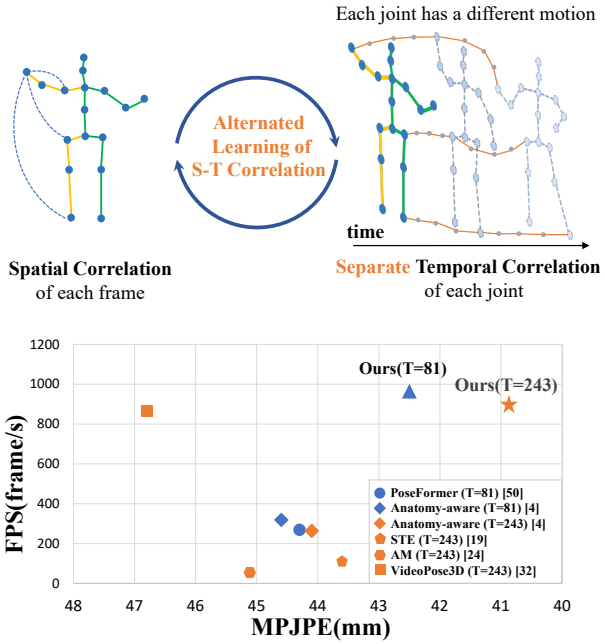


Figure 1. **Top:** Overview of spatio-temporal correlation modeling. Each 2D keypoint is separated in the temporal domain to learn different motion trajectories of body joints, and the spatial and temporal correlation are alternately stacked to improve the sequence coherence modeling ability. **Bottom:** Accuracy (MPJPE) and efficiency (FPS) comparison with different methods on Human3.6M dataset, the blue and orange colors indicate that the input sequence length T is equal to 81 and 243, respectively.

Notable progress has been made by exploiting temporal information contained in the input video to address the above issues in a single frame [1, 4, 16, 28, 37, 46]. Recently, driven by the success of transformer [45] for its ability to model sequence data, Zheng *et al*. [57] introduces a transformer-based 3D human pose estimation network. It takes advantage of spatio-temporal information for estimating the more accurate central-frame pose in video. By modeling spatial correlations between all joints and temporal

*Corresponding author: tuzhigang@whu.edu.cn
†Work done at Wuhan University

correlations among consecutive frames, PoseFormer [57] achieves performance improvement. However, it ignores the motion differences among body joints, which causes the insufficient learning of spatio-temporal correlation. Moreover, it increases the dimension of the temporal transformer module, which limits the usage of longer input sequence.

Poseformer [57] takes a video as input and only estimates the human pose of the central frame, which we summarize this pipeline as the *seq2frame* approach. Many recent methods [1,4,28,37,57] follow it and they utilize adjacent frames to improve the accuracy of estimating the pose of a certain moment, but the sequence coherence is ignored due to the single frame output. Additionally, during the inference, these *seq2frame* solutions need to input a 2D keypoint sequence repeatedly with large overlap to obtain 3D poses of all frames, which brings redundant calculation. In contrast to the *seq2frame* approach, there is also the *seq2seq* approach, which regresses the 3D pose sequence from the input 2D keypoints. These methods [16,46] mainly depend on long short-term memory (LSTM) [15] cell or graph convolution network (GCN) [21], and perform well in learning temporal information among continuous estimation results. However, current *seq2seq* networks lack the global modeling ability between input and output sequences, which tend to be excessively smooth [37] in the output poses of a long sequence. The low efficiency of LSTM [15] is also a severe issue for estimating human pose from video.

While previous work has focused on associating all joints in the spatial and temporal domains, we observe that the motion trajectories of the different body joints vary from frame to frame and should be learned separately. Additionally, the input 2D keypoint sequence and the output 3D pose sequence have solid global coherence, and they should be tightly coupled to promote accurate and smooth 3D poses.

Motivated by the above observations, in this work, we propose MixSTE to learn the separate temporal motion of each body joint and imbue sequential coherent human pose sequence in a *seq2seq* approach. In contrast to the prior method [57] which reconstructs the central frame and ignores the single joint motion, the MixSTE lifts 2D keypoint sequence to 3D pose sequence via a novel *seq2seq* architecture and a set of motion-aware constraints. Specifically, as shown at the top of Figure 1, we propose the joint separation to consider temporal motion information of each joint. It takes each 2D joint as an individual feature (which is referred to as a token in transformer) to sufficiently learn spatio-temporal correlation and helps to reduce the dimension of the joint features in temporal domain. Moreover, we propose an alternating design with *seq2seq* to flexibly obtain better sequence coherence within a long sequence, which decreases redundant calculation and excessive smoothness. In this way, temporal motion trajectories of different body joints could be adequately considered to predict accurate 3D pose sequence. To the best of our knowledge, the proposed method is the first to utilize the transformer encoder in the *seq2seq* pipeline, which enhances learning spatio-temperal correlation for accurate pose estimation and significantly improves the inference speed from seq2frame methods (see the bottom of Fig.1) Besides, our approach can easily adapt to any length of the input sequence.

Our contributions to 3D human pose estimation can be summarized in three folds:

- The MixSTE is proposed to effectively capture the temporal motion of different body joints over the long sequence, which helps to model sufficient spatio-temporal correlation.
- We propose a novel alternating design with transformer-based *seq2seq* model to learn the global coherence between sequences to improve the accuracy of reconstruction poses.
- Our approach achieves state-of-the-art performance on three benchmarks and has outstanding generalization.

2. Related Work

3D Human Pose Estimation. Estimating 3D human pose from monocular data was started by relying on the kinematics feature or the skeleton structure prior [17, 18, 38, 39]. With the development of deep learning, more data-driven methods have been proposed, and these methods can be divided into end-to-end manner and 2D-to-3D lifting manner. The end-to-end manner directly estimates the 3D coordinates from the input without the intermediate 2D pose representation. Some methods [36, 42, 44] followed this manner but required a high computation cost due to regressing directly from the image space. Different from the end-to-end manner, 2D-to-3D lifting pipeline first estimates 2D keypoints in the RGB data and then leverages the correspondences between 2D and 3D human structures to lift the 2D keypoints to 3D pose. Benefiting from the reliable effort of 2D keypoint detection works [8,13,29,34,41], recent 2D-to-3D lifting methods [9,27,30,31,48,56,58] outperformed end-to-end approaches. Therefore, we follow the 2D-to-3D lifting manner to obtain robust 2D intermediate supervision.

***Seq2frame* and *Seq2seq* under 2D-to-3D Lifting.** Recently, temporal information from video has been exploited to produce more robust predictions by many methods. With the video input, many influential works (*seq2frame*) pay attention to predicting the central frame of the input video to produce a more robust prediction and less sensitivity to noise. Pavllo *et al*. [37] proposed the dilated temporal convolutions based on the temporal convolution network (TCN) to extract temporal features. Some following works improved the performance of TCN by utilizing the attention

mechanism [28], or decomposing the pose estimation task into bone length and bone direction prediction [4], but they have to fix the receptive field of the input sequence. In contrast to them, our approach is no need to preset the length of each input with respect to the convolution kernel or the sliding window size. Besides, GCN [21] was also applied to the task by [1] to learn multi-scale features of human and hand poses. These works achieved good performance; however, calculation redundancy is a common flaw of these methods.

On the other hand, some works (*seq2seq*) improve the coherence and efficiency of 3D pose estimation and reconstruct all frames of input sequence at once. LSTM [15] was introduced to estimate 3D poses in video from a set of 2D keypoints [26]. Hossain *et al*. [16] presented a temporal derivative loss function to ensure the temporal consistency over a sequence, but it faces the low computing efficiency issue. Wang *et al*. [46] exploited a GCN-based approach and designed a corresponding loss to model motion in both short temporal intervals and long temporal ranges, but it lacks global modeling ability of input sequence. In contrast to [16, 46], our method has the advantage of global modeling ability of each joint in the spatial and temporal domains. Besides, it enables parallel processes for frames and joints to address the low-efficiency issue of LSTM [15].

Self-attention and Transformer The transformer architecture with self-attention was firstly proposed by [45], and then was applied to various visual tasks, *e.g*. classification with visual transformer (ViT) [10], and detection with DETR [2]. For the human pose estimation task, [49] proposed the Transpose to estimate 2D pose from images. [25] presented a transformer framework for both human mesh recovery and pose estimation from a single image but ignored the temporal information in the video. Some researchers also explored the multi-view 3D human pose estimation scheme [14]. The stride transformer encoder [23] was introduced to incorporate local contexts. Furthermore, PoseFormer [57] constructed a model based on ViT [10] to capture the spatial and temporal dependency sequentially. Both [23] and [57] have to fix the order of spatial and temporal encoders, and only the central frame of video is reconstructed. Our approach is similar to them in applying transformer architecture. But we consider motion trajectories of different body joints and apply the *seq2seq* to better model sequence coherence.

From the above analysis and comparison of related works, further exploration for transformer-based methods in 3D human pose estimation is necessary and feasible, but there is no method combining the transformer with *seq2seq* framework in the 3D human pose task.

3. Our Approach

As shown in Figure 2, our network takes a concatenated 2D coordinates $C_{N,T} \in \mathbb{R}^{N\times T\times 2}$ with N joints and T frames as input, where the channel size of the input is 2. Firstly, we project the input keypoint sequence $C_{N,T}$ to high-dimensional feature $P_{N,T} \in \mathbb{R}^{N\times T\times d_m}$ with feature dimension d_m for each joint representation. Then we utilize the position embedding matrix for retaining the position information of the spatial and temporal domains. The proposed MixSTE takes the $P_{N,T}$ as input and aims to alternately learn the spatial correlation and separate temporal motion. Finally, we use a regression head to concatenate the outputs $X \in \mathbb{R}^{N\times T\times d_m}$ of encoder, and take the dimension d_m to 3 to get the 3D human pose sequence $Out \in \mathbb{R}^{N\times T\times 3}$.

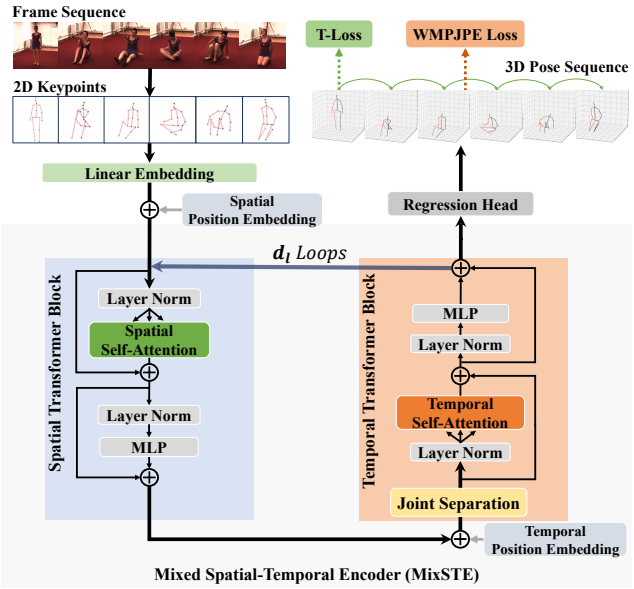


Figure 2. **Overview of the proposed framework.** The MixSTE is stacked for d_l loops, and each MixSTE models spatio-temporal dependencies independently. The WMPJPE Loss denotes the weighted per-joint position error loss. The T-Loss indicates the loss function of temporal coherence in Section 3.3.

3.1. Mixed Spatio-Temporal Encoder

We utilize the MixSTE to model spatial dependency and temporal motion for a given 2D input keypoint sequence, respectively. MixSTE consists of a Spatial Transformer Block (STB) and a Temporal Transformer Block (TTB). Here, the STB computes the self-attention between joints and aims to learn the body joint relations of each frame, while the TTB computes the self-attention between frames and focuses on learning the global temporal correlation of each joint.

3.1.1 Separate Temporal Correlation Learning

To imbue effective motion trajectories into the learned representations, we consider the temporal correspondence of each joint in order to explicitly model correlations on the same joint over the dynamic sequence. Different from the

previous method [57], we do not treat all body joints as a token in the temporal transformer block. We separate different joints in time dimension, so that the trajectory of each joint is an individual token $p \in \mathbb{R}^{1\times T\times d_m}$, and different joints of body are modeled paralleled. From the perspective of the time dimension, different motion trajectories of body joints are modeled separately to represent temporal correlations better. The joint separation is operated as follows:

$$X_l^t = Concat(\mathcal{F}(p_{i,1}, p_{i,2}, ...p_{i,T})),\ i \in N, \quad (1)$$

where $p_{i,j} \in P_{N,T}$ denotes the i-th joint in the j-th frame, $\mathcal{F}$ indicates the temporal encoder function and the output of the l-th TTB encoder is $X_l \in \mathbb{R}^{N\times T\times d_m}$. Furthermore, treating each body joint as an individual token can decrease dimension of the model to d_m from $N \times d_m$ of PoseFormer [57], and it also enables the longer sequence processed in the model.

3.1.2 Spatial Correlation Learning

We employ the spatial transformer block (STB) to learn spatial correlations among joints in each frame. Given 2D keypoints with N joints, we consider each joint as a token in spatial attention. Firstly, we take 2D keypoints as input and project each keypoint to a high-dimensional feature with the linear embedding layer. The feature is referred to as a spatial token in STB. We then embed the spatial position information with a positional matrix $E_{s-pos} \in \mathbb{R}^{N\times d_m}$. After that, spatial tokens $P_i \in \mathbb{R}^{N\times d_m}$ of the i-th frame is fed into spatial self-attention mechanism of STB to model dependencies across all joints and output the high-dimensional tokens $X_l^s \in \mathbb{R}^{N\times T\times d_m}$ in l-th STB.

3.1.3 Alternating design with *Seq2seq*

Alternating design in spatio-temporal correlation. The STB and TTB are designed in an alternating way to encode different high-dimensional tokens. The process of alternating design is like recurrent neural network (RNN), but we can parallel over joint and time dimensions. We stack STB and TTB for d_l loops, and the dimension of the feature is preserved as a fixed size d_m to promise that spatial-temporal correlation learning focuses on the same joint. Specifically, the spatial and temporal position embedding is applied only in the first encoder to retain two kinds of position information. Moreover, there is the independence of the spatial and temporal domains, where previous methods often only learn partial sequence coherence due to the single process of spatio-temporal modeling. The proposed alternating design with stacking architecture can obtain better coherence and spatio-temporal feature encoding.

***Seq2seq* framework.** Furthermore, to better utilize the global sequence coherence between the input sequence of 2D keypoints and the output sequence of 3D poses, we leverage the *seq2seq* pipeline in our model. It can predict all 3D poses of input 2D keypoints at once, which helps to preserve sequence coherence between the input and output sequences. Besides, for a sequence containing T frames, we need fewer times of inference, which means higher efficiency. Assuming that the sequence length of each input $t < T$, the inference time gap G between our model and the *seq2frame* methods will become higher with the increase of t:

$$G = \frac{T(1+2\delta)}{(\frac{T+2\delta}{t})} = \frac{T(1+2\delta)}{T+2\cdot\delta}\cdot t \approx (1+2\delta)\cdot t, \quad (2)$$

where δ indicates the padding length of the input sequence.

In summary, due to these advanced components, our model can capture various temporal motions and global sequence coherence with less calculation redundancy.

3.2. Transformer Block in MixSTE

The transformer blocks in MixSTE follow the scaled dot-product attention [45]. The attention computing of query, key, and value matrix Q, K, V in each head are formulated by:

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_m}})V, \quad (3)$$

where $\{Q, K, V\} \in \mathbb{R}^{N\times d_m}$, N indicates the number of tokens, and d_m is the dimension of each token. The concatenated attention of h heads is defined as follows:

$$MSA = Concat(head1, ..., head_h)W^O, \quad (4)$$

$$head_i = Attention(Q_i, K_i, V_i), i \in h, \quad (5)$$

where the linear projection weight is $W^O \in \mathbb{R}^{d_m\times d_m}$. In the transformer encoder of our approach, each joint token $p \in P_N$ is projected from joint c_i of the 2D coordinates $C_N \in \mathbb{R}^{N\times 2}$. Joint token p is embedded with the position information by a matrix $E_{pos} \in \mathbb{R}^{N\times d_m}$:

$$X = Norm(L_e(c_i) + E_{pos}),\ X \in \mathbb{R}^{N\times d_m}, \quad (6)$$

where $Norm$ denotes the layer normalization, and L_e indicates the linear embedding layer. The spatial-temporal dependencies among joints are then computed by the STB and TTB as follows:

$$R_s = MSA(U_Q, U_K, U_V) + X, \quad (7)$$

$$U_i = XW^m,\ m \in \{Q, K, V\}, \quad (8)$$

where R_s denotes the attention output of the joint token X, U_i is the matrix mapped from X by linear transformation, and W^m is the corresponding linear transformation weight matrix of query, key and value in joints.

3.3. Loss Function

The network is trained in an end-to-end manner, the final loss function $\mathcal{L}$ is defined as:

$$\mathcal{L} = \mathcal{L}_w + \lambda_t \mathcal{L}_t + \lambda_m \mathcal{L}_m, \tag{9}$$

where $\mathcal{L}_w$ is the WMPJPE loss, $\mathcal{L}_t$ is the TCLoss, and $\mathcal{L}_m$ denotes the MPJVE loss. During the training stage, different coefficients λ_t and λ_m are employed to $\mathcal{L}_t$ and $\mathcal{L}_m$ to avoid excessive smoothness in sequence.

In detail, we firstly explored a weighted mean per-joint position error (WMPJPE), which pays different attention to different joints of the human body when computing the MPJPE. The WMPJPE L_w with weight W is computed as follows:

$$\mathcal{L}_w = \frac{1}{N^s} \sum_{i=1}^{N^s} (W \times \frac{1}{T} \sum_{j=1}^{T} \parallel p_{i,j} - gt_{i,j} \parallel_2^2)), \tag{10}$$

where N^s indicates N joints of human skeleton s in three datasets, T denotes the number of frames in sequence, $p_{i,j}$ and $gt_{i,j}$ are the prediction and the ground truth 3D pose of i-th joint in j-th frame.

Moreover, the temporal consistency loss (TCLoss) in [16] is introduced to produce the smooth poses. The MPJVE [37] is also a loss in our model to improve the temporal coherence between the predicted pose sequence and the ground truth sequence. We merge the TCLoss and MPJVE as the temporal loss function (T-Loss).

4. Experiment

4.1. Datasets and Evaluation Protocols

We evaluate our model on three 3D human pose estimation datasets: Human3.6M [3,19], MPI-INF-3DHP [32] and HumanEva [40] individually.

Human3.6M is the most commonly used indoor dataset for the 3D human pose estimation tasks. Following the same policy of previous methods [4, 28, 31, 35–37, 57], the 3D human pose in Human3.6M is adopted as a 17-joint skeleton, and the subjects *S1, S5, S6, S7, S8* from the dataset are applied during training, the subjects *S9* and *S11* are used for testing. The two commonly used evaluation metrics (MPJPE and P-MPJPE) are involved in this dataset. In addition, mean per-joint velocity error (MPJVE) [37] is applied to measure the smoothness of the prediction sequence. We also compute the variance (VAR.) of MPJPE between action categories to evaluate the stability.

MPI-INF-3DHP is also a recently popular large-scale 3D human pose dataset. Our setting follows previous works [46, 57]. The area under the curve (AUC), percentage of correct keypoints (PCK), and MPJPE are reported as evaluation metrics.

HumanEva is a smaller dataset than above datasets. As the same setting of [28, 57], actions (Walk, Jog) in subjects *S1, S2, S3* are evaluation data. The metrics MPJPE and P-MPJPE are applied.

4.2. Implementation Details

The proposed model is implemented with Pytorch. We use 2D keypoints from 2D pose detector [8, 41] or 2D ground truth to analyze the performance of our framework. Although the proposed model can easily adapt to any length of input sequence, to be fair, we select some specific sequence lengths T for three datasets to compare our method with other methods which must have a certain 2D input length [4,28,37]: Human3.6M (T=81,243), MPI-INF-3DHP (T=1,27), HumanEva (T=81). Analysis about the frame length setting is discussed in the ablation study Section 4.4. The W in WMPJPE is set based on different joint groups (torso, head, middle limb, and terminal limb) with different values (1.0, 1.5, 2.5, and 4.0, respectively). The Adam optimizer [20] is employed for the training model. The batch size, dropout rate, and activation function for datasets are set to 1024, 0.1, and GELU. We utilize the stride data sample strategy with interval is as same as the input length to make there no overlapping frames between sequences(more details in the supplementary material).

4.3. Comparison with State-of-the-art Methods

Results on Human3.6M. Two types of 2D joint detection data are applied in the experiment: CPN [8], which is the most typical 2D estimator used in previous approaches, and HRNet [41] which is used to further investigate the upper bound of our method. The results compared with other methods, including the error of all 15 actions and the average error, are reported in Table 1. For CPN [8] detector, our model obtains the best result of average MPJPE of 40.9mm under Protocol 1 and 32.6mm P-MPJPE under Protocol 2, which outperforms PoseFormer [57] by 3.4mm MPJPE (**7.6%**). Furthermore, our method achieves the best under $T = 243$ setting and second-best under $T = 81$ setting in all actions.

Utilizing more powerful 2D detector HRNet [41], our model further improves roughly 4.5mm (**10.2%**) under Protocol 1. We also compare our method with [4,28,37,46,57] using 2D ground truth, and the results are illustrated in the Table 2. Our method significantly outperforms all other methods and achieves approximately **31.0%** improvement of average MPJPE compared with PoseFormer [57].

Furthermore, we compare the MPJPE distribution in the testset *S9* and *S11* with other methods [37, 57] to evaluate the ability of estimating difficult poses. It can be observed in Figure 3 that there are much fewer poses with high errors in our method. Moreover, the proportion of poses with over 40mm MPJPE, which causes loss of accuracy, is consis-

Protocol #1		Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg.**
Pavlakos *et al.* [35]	CVPR2018	48.5	54.4	54.4	52.0	59.4	65.3	49.9	52.9	65.8	71.1	56.6	52.9	60.9	44.7	47.8	56.2
Pavllo *et al.* [37](CPN, T=243)(†)	CVPR2019	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
Cai *et al.* [1](CPN, T=7)(†)	ICCV2019	44.6	47.4	45.6	48.8	50.8	59.0	47.2	43.9	57.9	61.9	49.7	46.6	51.3	37.1	39.4	48.8
Yeh *et al.* [51](†)	NIPS2019	44.8	46.1	43.3	46.4	49.0	55.2	44.6	44.0	58.3	62.7	47.1	43.9	48.6	32.7	33.3	46.7
Liu *et al.* [28](CPN, T=243)(†)	CVPR2020	41.8	44.8	41.1	44.9	47.4	54.1	43.4	42.2	56.2	63.6	45.3	43.5	45.3	31.3	32.2	45.1
Wang *et al.* [46](CPN, T=96)(†)	ECCV2020	40.2	42.5	42.6	41.1	46.7	56.7	41.4	42.3	56.2	60.4	46.3	42.2	46.2	31.7	31.0	44.5
Chen *et al.* [4](CPN, T=243)(†)	TCSVT2021	41.4	43.5	40.1	42.9	46.6	51.9	41.7	42.3	53.9	60.2	45.4	41.7	46.0	31.5	32.7	44.1
Xu *et al.* [48](T=1)	CVPR2021	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
Lin *et al.* [25](T=1)(*)	CVPR2021	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	54.0
Zeng *et al.* [53](†)	ICCV2021	43.1	50.4	43.9	45.3	46.1	57.0	46.3	47.6	56.3	61.5	47.7	47.4	53.5	35.4	37.3	47.9
Zheng *et al.* [57](CPN, T=81)(†)(*)	ICCV2021	41.5	44.8	39.8	42.5	46.5	51.6	42.1	42.0	53.3	60.7	45.5	43.3	46.1	31.8	32.2	44.3
Ours(CPN, T=81)(†)(*)		39.8	43.0	38.6	40.1	43.4	50.6	40.6	41.4	52.2	56.7	43.8	40.8	43.9	29.4	30.3	42.4
Ours(CPN, T=243)(†)(*)		**37.6**	**40.9**	**37.3**	**39.7**	**42.3**	**49.9**	**40.1**	**39.8**	**51.7**	**55.0**	**42.1**	**39.8**	**41.0**	**27.9**	**27.9**	**40.9**
Wang *et al.* [46](HRNet, T=96)(†)	ECCV2020	38.2	41.0	45.9	39.7	41.4	51.4	41.6	41.4	52.0	57.4	41.8	44.4	41.6	33.1	**30.0**	42.6
Wehrbein *et al.* [47](HRNet, T=200)	ICCV2021	38.5	42.5	39.9	41.7	46.5	51.6	39.9	40.8	49.5	56.8	45.3	46.4	46.8	37.8	40.4	44.3
Ours(HRNet, T=243)		**36.7**	**39.0**	**36.5**	**39.4**	**40.2**	**44.9**	**39.8**	**36.9**	**47.9**	**54.8**	**39.6**	**37.8**	**39.3**	**29.7**	30.6	**39.8**
Protocol #2		Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg.**
Wang *et al.* [46](CPN, T=96)(†)	ECCV2020	31.8	34.3	35.4	33.5	35.4	41.7	**31.1**	31.6	44.4	49.0	36.4	32.2	35.0	24.9	23.0	34.5
Liu *et al.* [28](CPN, T=243)(†)	CVPR2020	32.3	35.2	33.3	35.8	35.9	41.5	33.2	32.7	44.6	50.9	37.0	32.4	37.0	25.2	27.2	35.6
Zheng *et al.* [57](CPN, T=81)(†)(*)	ICCV2021	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
Ours(CPN, T=81)(†)(*)		32.0	34.2	31.7	33.7	34.4	39.2	32.0	31.8	42.9	46.9	35.5	32.0	34.4	23.6	25.2	33.9
Ours(CPN, T=243)(†)(*)		**30.8**	**33.1**	**30.3**	**31.8**	**33.1**	**39.1**	**31.1**	**30.5**	**42.5**	**44.5**	**34.0**	**30.8**	**32.7**	**22.1**	**22.9**	**32.6**
Wang *et al.* [46](HRNet)(†)	ECCV2020	28.4	32.5	34.4	32.3	32.5	40.9	30.4	29.3	42.6	**45.2**	33.0	32.0	33.2	24.2	22.9	32.7
Wehrbein *et al.* [47](HRNet, T=200)	ICCV2021	**27.9**	31.4	29.7	**30.2**	34.9	37.1	**27.3**	28.2	39.0	46.1	34.2	32.3	33.6	26.1	27.5	32.4
Ours(HRNet, T=243)		28.0	**30.9**	**28.6**	30.7	**30.4**	**34.6**	28.6	**28.1**	**37.1**	47.3	**30.5**	**29.7**	**30.5**	**21.6**	**20.0**	**30.6**
MPJVE		Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg.**
Pavllo *et al.* [37](†)	CVPR2019	3.0	3.1	2.2	3.4	2.3	2.7	2.7	3.1	2.1	2.9	2.3	2.4	3.7	3.1	2.8	2.8
Chen *et al.* [4](†)	TCSVT2021	2.7	2.8	**2.0**	3.1	2.0	2.4	2.4	2.8	1.8	2.4	2.0	**2.1**	3.4	**2.7**	2.4	2.5
Zheng *et al.* [57](†)(*)	ICCV2021	3.2	3.4	2.6	3.6	2.6	3.0	2.9	3.2	2.6	3.3	2.7	2.7	3.8	3.2	2.9	3.1
Ours(CPN, T=243)(†)(*)		**2.5**	**2.7**	**1.9**	**2.8**	**1.9**	**2.2**	**2.3**	**2.6**	**1.6**	**2.2**	**1.9**	**2.0**	**3.1**	**2.6**	**2.2**	**2.3**

Table 1. Detailed quantitative comparison results of MPJPE in millimeters (mm) on Human3.6M under Protocol 1 (no rigid alignment applied) and Protocol 2 (rigid alignment). **Top table**: results under Protocol 1 (MPJPE); **Middle table**: results under Protocol 2 (P-MPJPE); **Bottom table**: results of MPJVE. T denotes the number of input frames estimated by the respective approaches, (†) indicates using temporal information, and (*) indicates the transformer-based methods. The best and second-best results are highlighted in bold and underlined formats, respectively.

Protocol #1		Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg.**
Liu *et al.* [28](T=243)(†)	CVPR2020	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
Wang *et al.* [46](GT, T=96)	ECCV2020	23.0	25.7	22.8	22.6	24.1	30.6	24.9	24.5	31.1	35.0	25.6	24.3	25.1	19.8	18.4	25.6
Zheng *et al.* [57](T = 81)(†)(*)	ICCV2021	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
Ours(T=81)		25.6	27.8	24.5	25.7	24.9	29.9	28.6	27.4	29.9	29.0	26.1	25.0	25.2	18.7	19.9	25.9
Ours(T=243)		**21.6**	**22.0**	**20.4**	**21.0**	**20.8**	**24.3**	**24.7**	**21.9**	**26.9**	**24.9**	**21.2**	**21.5**	**20.8**	**14.7**	**15.7**	**21.6**

Table 2. Detailed quantitative comparison results of MPJPE in millimeters (mm) on Human3.6M under Protocol 1 using 2D ground truth keypoints as input. The best results are highlighted in bold.

tently lower, and the proportion of less than 30mm MPJPE is much higher than other methods. The results demonstrate our method performs better on difficult actions.

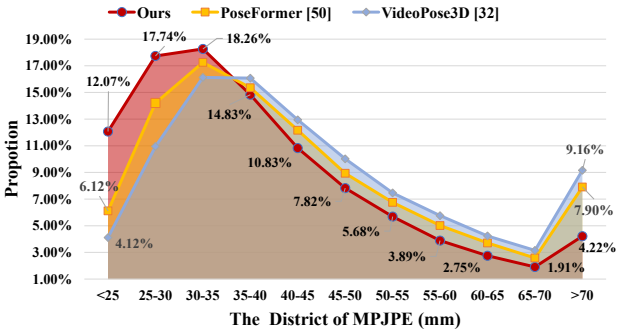


Figure 3. The MPJPE distribution on Human3.6M testset.

In Figure 4, we compare the MPJPE for individual joints on all frames of Human3.6M testset to evaluate the estimation accuracy of different joints. The joints of limbs have higher errors due to flexible movements, while the trunk joints have lower errors because of stable motion. Our accuracy of each joint category achieves the best, and the variance ($VAR.$) comparison shows that our method has a more stable performance.

Results on MPI-INF-3DHP. Table 3 reports the detailed comparison with other methods on the MPI-INF-3DHP testset. In addition, the 1-frame setting is employed to evaluate the single-frame performance. The input is ground truth 2D keypoints. As shown in the table, the method (T=27) performs the best in three evaluation metrics, and the single-frame setting (T=1) also achieves the second-best accuracy. These results demonstrate the strong performance of our model in single-frame and multi-frame scenarios.

Reults on HumanEva. We utilize HumanEva to evaluate the generalization ability of the proposed method and

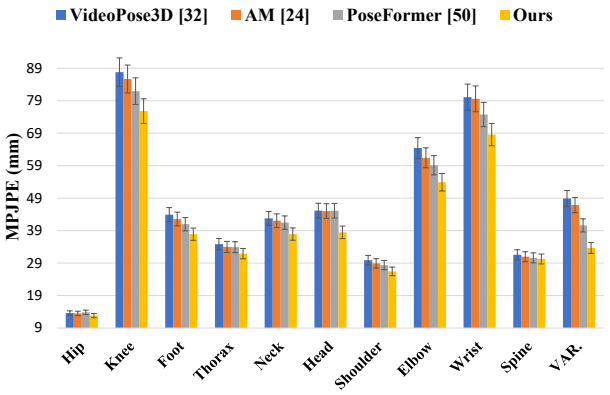


Figure 4. The average joint error comparison across all frames of the testset in the Human3.6M. The $VAR.$ indicates the variance among joint errors divided by a factor (10.0), and the joints of the same part (*e.g.* right knee and left knee) are divided into the same category for the sake of display.

Method		PCK↑	AUC↑	MPJPE↓
Mehta *et al.* [33]	ACM TOG 2017	79.4	41.6	-
Lin *et al.* [24](T=25)	BMVC2019	83.6	51.4	79.8
Li *et al.* [22]	CVPR2020	81.2	46.1	99.7
Wang *et al.* [46](T=96)	ECCV2020	86.9	62.1	68.1
Chen *et al.* [4](T=243)	TCSVT2021	87.8	53.8	79.1
Gong *et al.* [12]	CVPR2021	88.6	57.3	73.0
Zheng *et al.* [57]	ICCV2021	88.6	56.4	77.1
Ours(T=1)		<u>94.2</u>	<u>63.8</u>	<u>57.9</u>
Ours(T=27)		**94.4**	**66.5**	**54.9**

Table 3. Detailed quantitative comparison results on MPI-INF-3DHP with three metrics. The ↑ indicates the higher, the better, the ↓ indicates the lower, the better. The best and second-best results are highlighted in bold and underlined formats, respectively.

the impact of finetuning from large datasets. The MPJPE results on HumanEva finetuning from Human3.6M are reported in the Table 4. Due to *seq2seq* setting and limitation of transformer in small dataset, our method without finetuning is slightly worse than our baseline. But the performance can be improved by using smaller data sample strides (interval=1). The experiment shows that our model has a better generalization ability than previous methods.

#Protocol1	Walk			Jog			**Avg.**
Pavllo *et al.* [37](T=81)	13.1	**10.1**	39.8	20.7	13.9	15.6	18.9
Pavllo *et al.* [37](T=81, FT)	14.0	12.5	27.1	**20.3**	17.9	17.5	18.2
Zheng *et al.* [57](T=43)	16.3	11	47.1	25	15.2	15.1	21.6
Zheng *et al.* [57](T=43, FT)	14.4	10.2	46.6	22.7	**13.4**	**13.4**	20.1
Ours(T=43)	20.3	22.4	34.8	27.3	32.1	34.3	28.5
Ours(T=43, interval=1)	16.2	14.2	21.6	24.6	23.2	25.8	20.9
Ours(T=43, FT)	**12.7**	10.9	**17.6**	22.6	15.8	17.0	**16.1**

Table 4. The MPJPE on HumanEva testset under Protocol 1. FT indicates using the pretrained model on Human3.6M for finetuning. The best result is highlighted in bold.

4.4. Ablation Study

To evaluate the impact and performance of each component in our model, we evaluate their effectiveness in this section. The Human3.6M dataset and the CPN [8] detector are employed to provide 2D keypoints.

Effect of Each Component. As shown in Table 5, we first modify the central frame 3D pose output to the sequence output without any other optimization to get the *seq2seq* baseline model. For a fair comparison, the parameter setting of the *seq2seq* baseline is directly applied to the proposed method, and the MPJPE loss is utilized in the baseline model. After applying the alternating design, the result shows that our method decreases 6.2mm MPJPE (from 51.7mm to 45.5mm). Then joint separation is utilized to demonstrate its advantage in both improving the performance (from 45.5 to 41.7) and reducing computing cost (FLOPs for each frame decreases to 645 from 186405). By applying our loss function to replace MPJPE loss, our result achieves the best (40.9mm MPJPE with 645 FLOPs). The MixSTE with our loss function improves 20.9% (from 51.7 to 40.9) compared to the *seq2seq* baseline, and it proves the rationality of our network design.

Effect of Loss Function. We have explored the contribution of our loss function in detail. As shown in Table 6, the MPJPE metric decreases from 41.7 to 41.3 after applying the WMPJPE loss. The result demonstrates that the WMPJPE is an essential loss to improve accuracy. Then the temporal consistency loss (TCLoss) following [16] is employed to improve the temporal smoothness performance (MPJVE) by 1.0 (decreases from 4.6 to 3.6), and the coherence gets better after using the MPJVE loss (decreases from 4.6 to 2.6). The motion loss [46] has less contribution to the coherence than TCLoss and MPJVE loss. Finally, after applying the T-Loss and WMPJPE loss to our method, the result achieves the best on the MPJPE and MPJVE metrics

	Seq2seq	Alternating Design	Joint Separation	Our Loss	MPJPE	FLOPs (M)
Baseline	✓				51.7	186405
	✓	✓			45.5	186405
	✓	✓	✓		41.7	645
Ours	✓	✓	✓	✓	40.9	645

Table 5. Ablation study for each component used in our method. The evaluation is performed on Human3.6M with MPJPE (mm) and FLOPs.

	MPJPE	MPJVE
MPJPE Loss	41.7	5.0
WMPJPE Loss	41.3	4.6
WMPJPE Loss + Motion Loss [46]	41.3	4.3
WMPJPE Loss + TCLoss [16]	41.2	3.6
WMPJPE Loss + MPJVE Loss	41.2	2.6
Ours (WMPJPE Loss + T-Loss)	40.9	2.3

Table 6. Ablation study for loss function in our method with MPJPE and MPJVE.

(40.9mm MPJPE, 2.3 MPJVE). The ablation study demonstrates that our loss function is comprehensive for the proposed model regarding accuracy and smoothness.

Parameter Setting Analysis. Table 7 shows how the setting of different hyper-parameters in our method impacts the performance under Protocol 1 with MPJPE. There are three main hyper-parameters for the network: the depth of MixSTE (d_l), the dimension of model (d_m), and the input sequence length (T). We divide the configurations into 3 groups row-wise, and different values are assigned for one hyper-parameters while keeping the other two hyper-parameters fixed to evaluate the impact and choice of each configuration. Based on the results in the table, we choose the combination of $Depth$=8, $Channel$=512, and $Input\ Length$=243. Note that we choose the $Depth = 8$ rather than $Depth = 10$ because the latter setting introduces a more significant number of parameters (33.7M *vs.* 42.2M).

Depth (d_l)	Dimension (d_m)	Input Length (T)	MPJPE
4	64	27	54.3
6	64	27	53.2
8	64	27	51.8
10	64	27	51.1
8	128	27	47.9
8	256	27	46.1
8	**512**	27	45.1
8	640	27	46.0
8	512	81	42.7
8	512	128	42.0
8	512	**243**	**40.9**
8	512	300	41.8

Table 7. Ablation study for hyper-parameter setting in depth (d_l), dimension (d_m) and input length (T). The evaluation is performed on Human3.6M with MPJPE (mm).

4.5. Qualitative Results

As shown in Figure 5, we further conduct visualization on spatial and temporal attention. The selected action (*SittingDown* of testset *S11*) is applied for visualization. Moreover, attention outputs of different heads are averaged to observe the overall correlations of joints and frames, and the attention outputs are normalized to $[0, 1]$. It can be easily observed from spatial attention map (left of Figure 5) that our model learns different dependencies between joints. Furthermore, we also visualize the temporal attention map (right of Figure 5) from the last temporal attention layer. The two parts with light color have similar poses with adjacent frames, while the dark color corresponded frame (the middle image in the frame sequence) has a more different pose with adjacent frames. We also evaluate the visual result of estimated poses and 3D ground truth of Human3.6M in Figure 6 to show that we can estimate more accurate poses compared to PoseFormer [57].

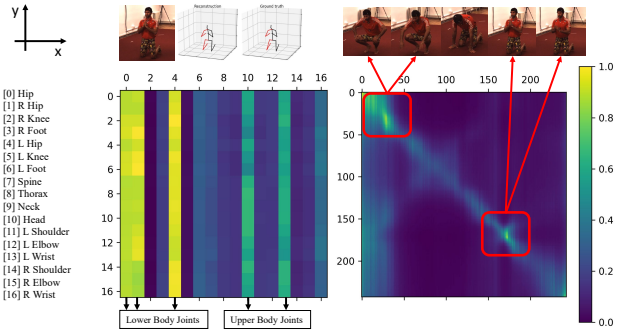


Figure 5. Visualization of self-attentions among body joints and frames. The x-axis and y-axis correspond to the queries and the predicted outputs, respectively. Each row shows the attention weight $w_{i,j}$ of the j-th query for the i-th output.

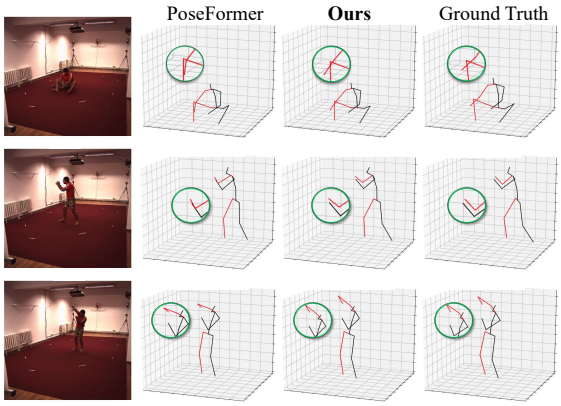


Figure 6. Qualitative comparison between our method (MixSTE) and [57] with the *Photo* and *SittingDown* actions on on Human3.6M. The green circle highlights locations where our method has better results.

5. Conclusion

We have presented MixSTE, a novel transformer-based *seq2seq* approach for 3D pose estimation from monocular video. The model can better capture global sequence coherence and temporal motion trajectories of different body joints. Moreover, the efficiency of 3D human pose estimation is much improved. Comprehensive evaluation results show that our model obtains the best performance. As a new universal baseline, the proposed method also opens up many possible directions for future works. Nonethless, our method is still limited by inaccurate 2D detection results *e.g.* missing and noisy keypoints. It may be alleviated by applying better 2D detector, but modeling distribution of input noise is also a feasible and valuable exploration.

Acknowledgements. This work was supported by the National Natural Science Foundation of China under Grant 62106177 and 61773272.

References

[1] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2272–2281, 2019. 1, 2, 3, 6

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 3

[3] Cristian Sminchisescu Catalin Ionescu, Fuxin Li. Latent structured models for human pose estimation. In *International Conference on Computer Vision*, 2011. 5

[4] Tianlang Chen, Chen Fang, Xiaohui Shen, Yiheng Zhu, Zhili Chen, and Jiebo Luo. Anatomy-aware 3d human pose estimation with bone-based pose decomposition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021. 1, 2, 3, 5, 6, 7

[5] Yujin Chen, Zhigang Tu, Liuhao Ge, Dejun Zhang, Ruizhi Chen, and Junsong Yuan. So-handnet: Self-organizing network for 3d hand pose estimation with semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6961–6970, 2019. 1

[6] Yujin Chen, Zhigang Tu, Di Kang, Linchao Bao, Ying Zhang, Xuefei Zhe, Ruizhi Chen, and Junsong Yuan. Model-based 3d hand reconstruction via self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10451–10460, 2021. 1

[7] Yujin Chen, Zhigang Tu, Di Kang, Ruizhi Chen, Linchao Bao, Zhengyou Zhang, and Junsong Yuan. Joint hand-object 3d reconstruction from a single image with cross-branch feature fusion. *IEEE Transactions on Image Processing*, 30:4008–4021, 2021. 1

[8] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7103–7112, 2018. 2, 5, 7

[9] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 3

[11] Jia Gong, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. Meta agent teaming active learning for pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1

[12] 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 (CVPR)*, pages 8575–8584, June 2021. 7

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[14] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2, 3

[16] 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)*, September 2018. 1, 2, 3, 5, 7

[17] Catalin Ionescu, Joao Carreira, and Cristian Sminchisescu. Iterated second-order label sensitive pooling for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2

[18] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, July 2014. 2

[19] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014. 5

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. 5

[21] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 2, 3

[22] Shichao Li, Lei Ke, Kevin Pratama, Yu-Wing Tai, Chi-Keung Tang, and Kwang-Ting Cheng. Cascaded deep monocular 3d human pose estimation with evolutionary training data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 7

[23] Wenhao Li, Hong Liu, Runwei Ding, Mengyuan Liu, Pichao Wang, and Wenming Yang. Exploiting temporal contexts with strided transformer for 3d human pose estimation. *IEEE Transactions on Multimedia*, 2022. 3

[24] Jiahao Lin and Gim Hee Lee. Trajectory space factorization for deep video-based 3d human pose estimation. *arXiv preprint arXiv:1908.08289*, 2019. 7

[25] 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*, pages 1954–1963, 2021. 3, 6

[26] Mude Lin, Liang Lin, Xiaodan Liang, Keze Wang, and Hui Cheng. Recurrent 3d pose sequence machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[27] Kenkun Liu, Rongqi Ding, Zhiming Zou, Le Wang, and Wei Tang. A comprehensive study of weight sharing in graph networks for 3d human pose estimation. In *European Conference on Computer Vision*, pages 318–334. Springer, 2020. 2

[28] Ruixu Liu, Ju Shen, He Wang, Chen Chen, Sen-ching Cheung, and Vijayan Asari. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5064–5073, 2020. 1, 2, 3, 5, 6

[29] Xianzheng Ma, Hossein Rahmani, Zhipeng Fan, Bin Yang, Jun Chen, and Jun Liu. Remote: Reinforced motion transformation network for semi-supervised 2d pose estimation in videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 2

[30] Xiaoxuan Ma, Jiajun Su, Chunyu Wang, Hai Ci, and Yizhou Wang. Context modeling in 3d human pose estimation: A unified perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6238–6247, 2021. 2

[31] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2640–2649, 2017. 1, 2, 5

[32] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 Fifth International Conference on*. IEEE, 2017. 5

[33] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 7

[34] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 2

[35] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 5, 6

[36] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 5

[37] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 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*, pages 7753–7762, 2019. 1, 2, 5, 6, 7

[38] Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Reconstructing 3D Human Pose from 2D Image Landmarks. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, Lecture Notes in Computer Science, pages 573–586, Berlin, Heidelberg, 2012. Springer. 2

[39] Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Reconstructing 3d human pose from 2d image landmarks. In *European conference on computer vision*, pages 573–586. Springer, 2012. 2

[40] 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. *International journal of computer vision*, 87(1-2):4, 2010. 5

[41] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 2, 5

[42] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2018. 2

[43] Mikael Svenstrup, Soren Tranberg, Hans Jorgen Andersen, and Thomas Bak. Pose estimation and adaptive robot behaviour for human-robot interaction. In *2009 IEEE International Conference on Robotics and Automation*, pages 3571–3576, 2009. 1

[44] Bugra Tekin, Artem Rozantsev, Vincent Lepetit, and Pascal Fua. Direct prediction of 3d body poses from motion compensated sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 991–1000, 2016. 2

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1, 3, 4

[46] Jingbo Wang, Sijie Yan, Yuanjun Xiong, and Dahua Lin. Motion guided 3d pose estimation from videos. In *European Conference on Computer Vision*, pages 764–780. Springer, 2020. 1, 2, 3, 5, 6, 7

[47] Tom Wehrbein, Marco Rudolph, Bodo Rosenhahn, and Bastian Wandt. Probabilistic monocular 3d human pose estimation with normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11199–11208, October 2021. 6

[48] 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*, pages 16105–16114, 2021. 2, 6

[49] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Towards explainable human pose estimation by transformer. *arXiv preprint arXiv:2012.14214*, 2020. 3

[50] Mang Ye, He Li, Bo Du, Jianbing Shen, Ling Shao, and Steven C. H. Hoi. Collaborative refining for person re-identification with label noise. *IEEE Transactions on Image Processing*, 31:379–391, 2022. 1

[51] Raymond Yeh, Yuan-Ting Hu, and Alexander Schwing. Chirality nets for human pose regression. *Advances in Neural Information Processing Systems*, 32:8163–8173, 2019. 6

[52] Jae Shin Yoon, Lingjie Liu, Vladislav Golyanik, Kripasindhu Sarkar, Hyun Soo Park, and Christian Theobalt. Pose-guided human animation from a single image in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15039–15048, June 2021. 1

[53] Ailing Zeng, Xiao Sun, Lei Yang, Nanxuan Zhao, Minhao Liu, and Qiang Xu. Learning skeletal graph neural networks for hard 3d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11436–11445, October 2021. 6

[54] Can Zhang, Tianyu Yang, Junwu Weng, Meng Cao, Jue Wang, and Zou Yuexian. Unsupervised pre-training for temporal action localization tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1

[55] Jiaxu Zhang, Gaoxiang Ye, Zhigang Tu, Yongtao Qin, Jinlu Zhang, Xiangjian Liu, and Shixu Luo. A spatial attentive and temporal dilated (satd) gcn for skeleton-based action recognition. *CAAI Transactions on Intelligence Technology*, 2020. 1

[56] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[57] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11656–11665, October 2021. 1, 2, 3, 4, 5, 6, 7, 8

[58] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 398–407, 2017. 2

Supplementary Material of MixSTE: Seq2seq Spatio-Temporal Encoder for 3D Human Pose Estimation in Video

Jinlu Zhang[1] Zhigang Tu[1*] Jianyu Yang[2] Yujin Chen[3†] Junsong Yuan[4]
[1]Wuhan University [2]Soochow University [3]Technical University of Munich
[4]State University of New York at Buffalo
{jinluzhang, tuzhigang}@whu.edu.cn, jyyang@suda.edu.cn, yujin.chen@tum.de, jsyuan@buffalo.edu

A. Implementation Details

Algorithm 1 shows our proposed MixSTE. We implement the proposed approach with Pytorch, and the model could support inference on a single NVIDIA GTX 2080Ti GPU. Each epoch takes about 22 minutes, and we train for about 160 epochs. The input 2D predicted keypoints of Human3.6M are estimated by the Cascaded Pyramid Network (CPN) [**?**] or HRNet [**?**]. The CPN detection result released by [**?**] is employed in experiments, and the HRNet detection result is acquired from fine-tuning pre-trained model to the Human3.6M dataset. The batch size of the HRNet is set to 64 for training, and the initial learning rate is 5e-4, using the step learning rate decaying policy. The final layer of the HRNet model is modified to learn to regress a set of a 17-joint skeleton.

Adam optimizer [**?**] is employed for the model training with the initial learning rate of 4e-5, using the exponential learning rate decay schedule (the multiplicative factor is set to 0.99, 0.99, and 0.995 for the Human3.6M, MPI-INF-3DHP, and HumanEva, respectively). Data augmentation is applied to training and test data by flipping the pose horizontally, following [**?**, **?**].

A stride data sample strategy is utilized to split the long sequence data during our training (also see analysis in Section C). We sample the 2D keypoints in a video with a stride step that is equal to the sequence length of the network input.

Algorithm 1: Mixed Spatio-Temporal Encoder Configuration

Input: Number of the stacked MixSTEs: L,
2D pose sequence: $P_{N,T} = \{p_{0,0}, ...p_{N,T}\}$,
Dimension of attention mecahnism: d.
Output: High-dimensional output F^l for each sequence

T, $N \leftarrow$ shape of $P_{N,T}$
$F^l_{N,T}$ = LinearProjection($P_{N,T}$)
for $l \leftarrow 0$ *to* $L-1$ **do**
  if $l = 0$ **then**
    $F^l_{N,T} \leftarrow$ Spatio-Temporal Position E_{pos}
  // Spatial Block
  $S^l_{0:N} \leftarrow$ Exact N dimension of $F_{N,T}$
  AS = **Spatial Attention** of $\{p_0, ...p_N\}$
  $S^l_{0:N} = S^l_{0,...N} + AS$
  $S^l_{0:N} = S^l_{0,...N} + MLP(S^l_{0:N})$
  $F^l_{0:N,T} \leftarrow S^l_{0:N}$
  // Temporal Block
  $T^l_{0:T} \leftarrow$ **Cross** N and T Dimension of $F^l_{N,T}$
  AT_i = **Temporal Attention** for each joint $\{p_{i,0}, ...p_{i,T}\}$
  AT = **Concat**($\{AT_0, ...AT_T\}$)
  $T^l_{0:T} = T^l_{0:T} + AT$
  $T^l_{0:T} = T^l_{0:T} + MLP(T^l_{0:T})$
  $F^l_{N,0:T} \leftarrow T^l_{0:T}$
return $F^{L-1}_{N,0:T}$

B. Loss Function Details

We apply multiple loss functions in the training stage to supervise the model training. Based on commonly-used mean per-joint position error (MPJPE), we use a weighted mean per-joint position error (WMPJPE) to re-weight different joints of the body. Larger weights are used for joints with drastic motion amplitudes. According to the amplitude of motion, all body joints are divided into three categories: the torso, the limb mid, and the limb end. The weight assigned to the torso is the smallest, and the weight assigned to the endpoints is the largest.

The WMPJPE $\mathcal{L}_w$ with weight w_i for i-th joint is computed as follows:

$$\mathcal{L}_w = \frac{1}{N}\sum_{i=1}^{N}(w_i \times MPE(p_i, gt_i)), \tag{1}$$

*Corresponding author
†Work done at Wuhan University

where N indicates N joints of skeleton, p and gt are the predicted and ground truth of 3D pose. We use MPE function to denote the mean position error (MPE) of the i-th joint in time dimension:

$$MPE(p_i, gt_i) = \frac{1}{T}\sum_{j=1}^{T} \parallel p_{j,i} - gt_{j,i} \parallel_2^2, \tag{2}$$

where T indicates the number of frames of sequence. The predicted and ground truth 3D pose in i-th frame are denoted as $p_{j,i}$ and $gt_{j,i}$. WMPJPE provides different supervision for each joint in space. Considering there is no much displacement of poses between adjacent frames, we follow the [**?**] and apply the L2 norm of the first derivative of WMPJPE in the time dimension to one of the loss functions in order to make the pose smooth in the time dimension. The temporal consistency loss (TCLoss) $\mathcal{L}_t$ is defined as:

$$\mathcal{L}_t = \frac{1}{NT}\sum_{j=2}^{T}\sum_{i=1}^{N} \parallel (p_{j,i} - p_{j-1,i}) \parallel_2^2, \tag{3}$$

where $p_{j,i}$ is the predicted location of the i-th joints in j-th frame.

The MPJVE $\mathcal{L}_m$ is also utilized in our model to improve the motion coherence [**?**] between the predicted poses and ground truth.

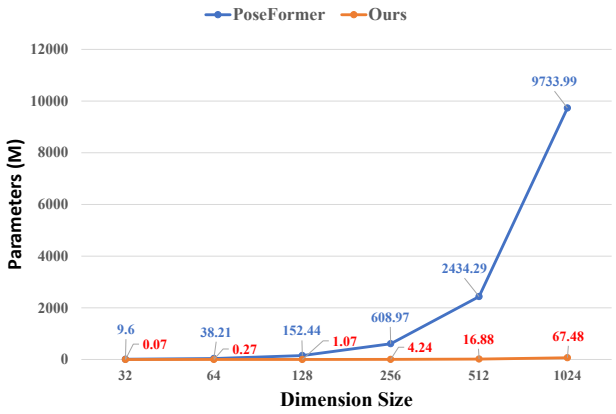
During the training stage, λ_t and λ_m are applied to weight $\mathcal{L}_t$ and $\mathcal{L}_m$. Therefore we train the network in an end-to-end manner with the multi loss function:

$$\mathcal{L} = \mathcal{L}_w + \lambda_t\mathcal{L}_t + \lambda_m\mathcal{L}_m. \tag{4}$$

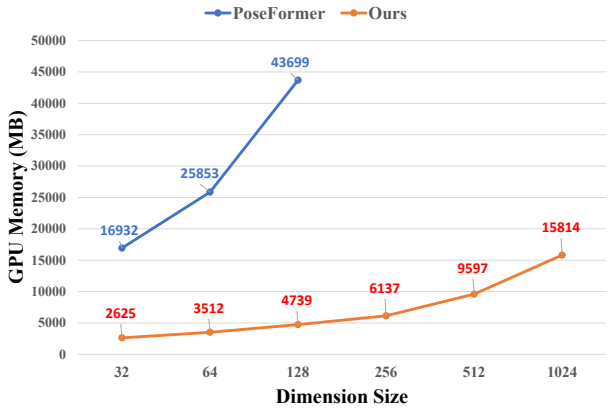
C. Additional Results

Comparison with PoseFormer. We compare the parameters, memory occupy, and training time per epoch of our model with PoseFormer [**?**]. For both our method and PoseFormer, we use 4 transformer encoders and set the input sequence length to be 243. When increasing the dimension of the self-attention block, we observe that PoseFormer requires more parameters, GPU memory, and running time of a training epoch than Ours (see Figure 7, showing our proposed MixSTE is more efficient.). As shown in the Table 8, the proposed method achieve better performance (lower MPJPE) with faster speed (higher FPS, lower FLOPs) than PoseFormer. The computing of FLOPs follows the [**?**, **?**].

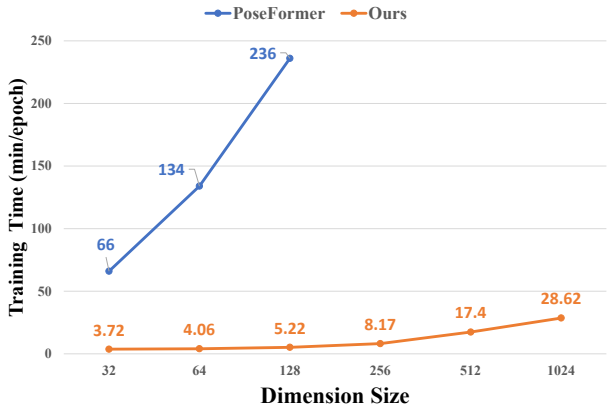
Effect of Data Sample Strategy. As shown in Figure 8, our stride data sample strategy results in fewer iterations to complete each training sample, thereby reducing overall training time. The stride data sample strategy is evaluated with different intervals. The max interval is equal to the input length, which means there is no overlap between frames. When $interval = 1$, the sampling is step by step. As shown in the Table 9, our method with max interval achieves best,



(a) The parameter size comparisons.



(b) The GPU memory occupy comparisons.



(c) The training time for each epoch comparisons.

Figure 7. Comparison of parameter, memory occupy, and training time with PoseFormer [**?**]. The dimension size is the dimension of each query, key, and value in the encoders, which is the main factor of model size.

which demonstrates that the strategy keeps the performance and successfully reduces the training time.

Discussion of Sparse Attention. To further explore the

Methods	FPS↑	FLOPs (M)↓	MPJPE↓
PoseFormer [?] (T=81)	288	1593	44.3
Ours (T=81)	965	965	42.5
Ours (T=243)	897	645	40.9

Table 8. Comparison with PoseFormer [?] in terms of frame per second (FPS), computating cost for each frame (FLOPs), and MPJPE. The evaluation is performed on Human3.6M testset *S9, S11* under Protocol 1 with CPN [?] as the 2D pose detector. Computation is done on a single GTX 2080Ti GPU.

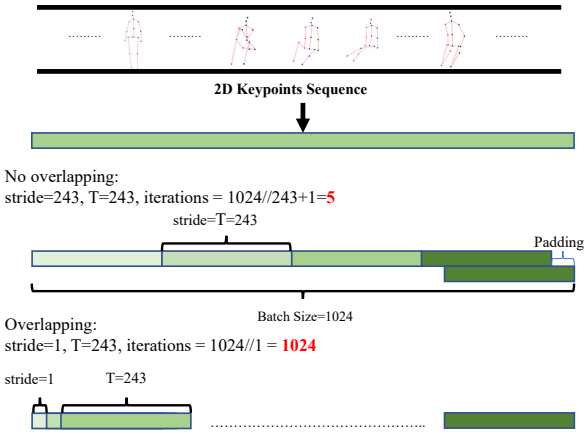


Figure 8. The processing example of stride data sample strategy. The stride example has fewer iterations than the example without stride sample, leading to less training time.

Input Length	Sample Strategy	MPJPE
27	Ours (interval=27)	54.3
27	interval=9	56.9
27	interval=3	67.3
27	interval=1	78.8

Table 9. Ablation studies on the data sample strategy on Human3.6M under Protocol 1 with MPJPE (mm). The input length is set to 27, and the intervals are 27, 9, 3, 1, respectively.

sparse attention for our proposed method, we experiment some recent sparse attention works [?, ?, ?, ?]. The result shown in the Figure 9 illustrates the different sparse attention prototypes can effectively converge in our framework and present similar convergence rates in training and testing. But there is still an accuracy gap compared with the full attention [?] used in our approach. Therefore, suitable sparse attention mechanism for our method could be one of the exploration directions in the future.

Qualitative Results of Attention Visualization. The qualitative results of all attention heads are also reported. We evalauate the proposed model on the Human3.6M dataset test set *S11* with *SittingDown* action. The spatial attention

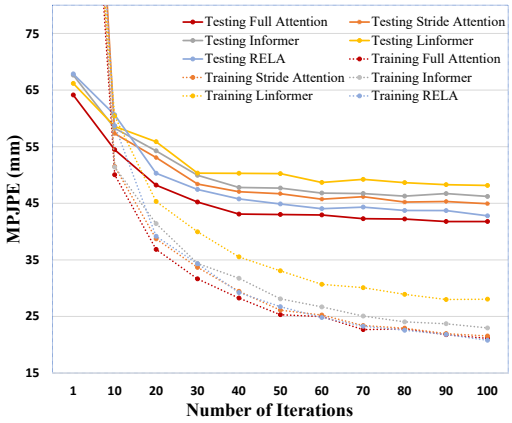


Figure 9. Comparison of different sparse attention and full attention mechanism for our method.

maps and temporal attention maps are shown in the Figure 10 and Figure 11, respectively. We can observe that attention heads have different intensities on body joints and frames, representing the local relationships modeled among the input sequence in each heads domain. The attention maps in the spatial domain tend to focus on some of the joints, and the maps in the temporal domain tend to have strong sensitivity over certain frames themselves. It illustrates that the feasibility of sparse attention in the temporal domain.

Qualitative Results of Inference in-the-Wild Video. Estimating the 3D human pose from in-the-wild videos is more challenging and meaningful. We apply CPN [?] as the 2D keypoints detector firstly, and then we utilize the MixSTE to obtain the 3D human pose. As shown in the Figure 12, our method achieves high robustness and accuracy in most of the frames of wild videos with challenging scenarios of occlusion and extremely fast motion.

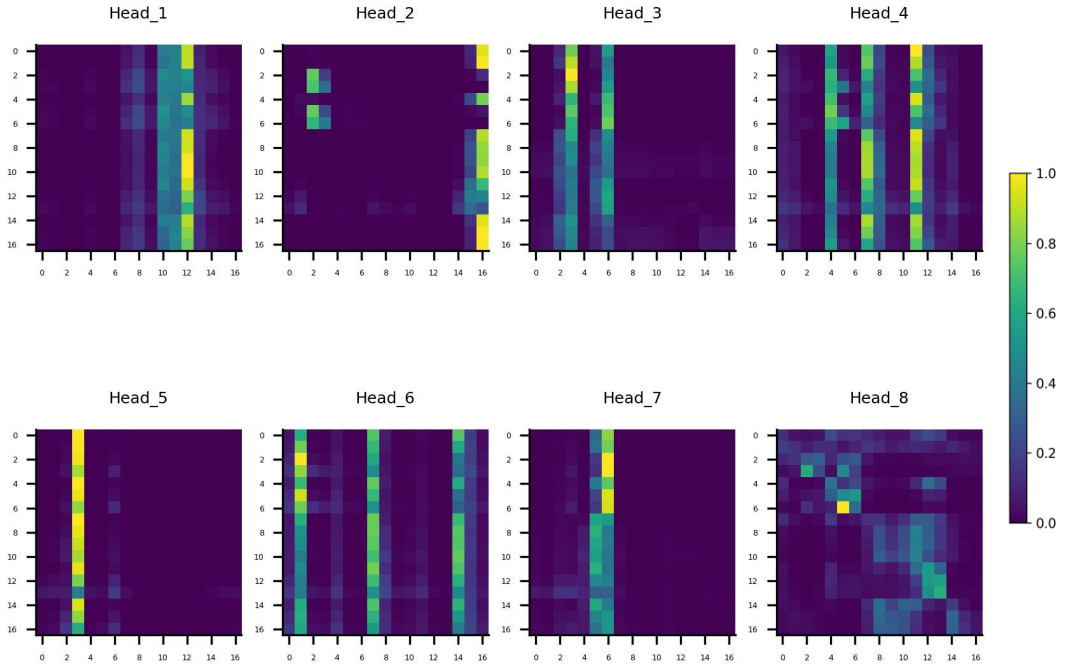


Figure 10. Qualitative Results of all heads attention maps among body joints. The x-axis (horizonal) and y-axis (vertical) to the joints queries and the predicted outputs, respectively. Each row shows the attention weight $w_{i,j}$ of the j-th query for the i-th output. The attention output is normalized from 0 to 1, and lighter color indicates stronger attention.

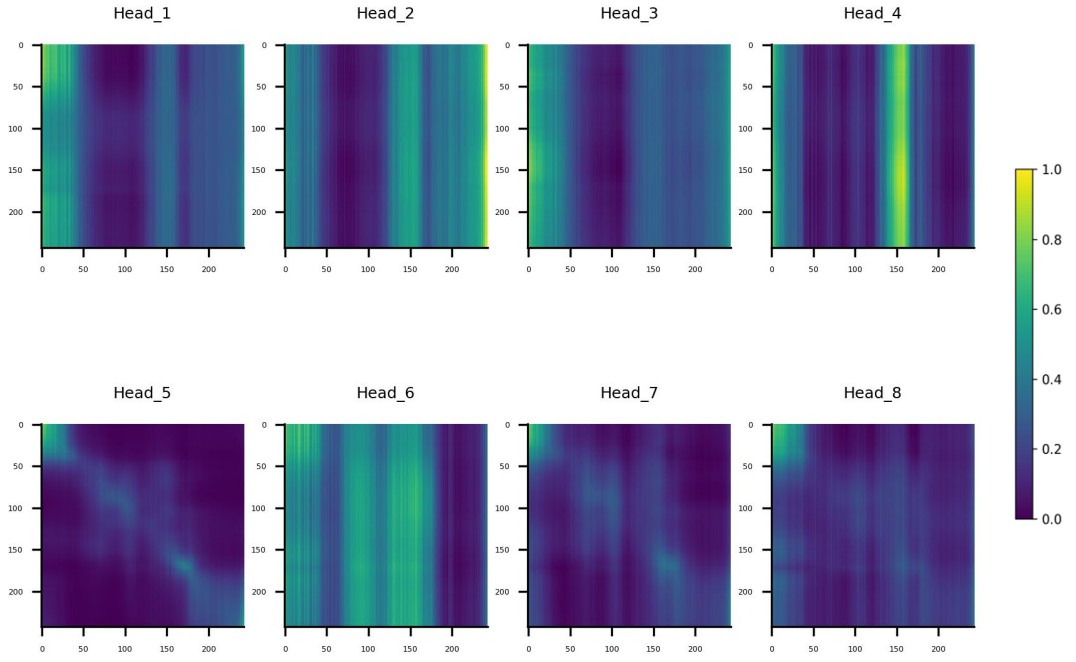


Figure 11. Qualitative Results of all heads attention maps among sequence frames. The x-axis (horizonal) and y-axis (vertical) correspond to the frames queries and the predicted outputs, respectively. Each row shows the attention weight $w_{i,j}$ of the j-th query for the i-th output. The attention output is normalized from 0 to 1, and lighter color indicates stronger attention.

Figure 12. Qualitative Results of in-the-wild video. The video frame sequences with detected 2D joints and corresponding recontructed 3D poses are shown.