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

Jinlu Zhang1 Zhigang Tu1* Jianyu Yang2 Yujin Chen3† Junsong Yuan4
1Wuhan University 2Soochow University 3Technical University of Munich 4State 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 Pytorvh, 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) [1] or HRNet [7]. The CPN detection result released by [5] 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 exponential learning rate decay schedule (the multiplicative factor is 0.99, 0.99, and 0.995 for the Human3.6M, MPI-INF-3DHPI, and HumanEva, respectively). Data augmentation is applied to training and test data by flipping the pose horizontally, following [5, 10].

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.

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 $L_w$ with weight $w_i$ for $i$-th joint is computed as follows:

$$L_w = \frac{1}{N} \sum_{i=1}^{N} (w_i \times MPE(p_i, gt_i)),$$

(1)

Algorithm 1: Mixed Spatio-Temporal Encoder

Input: Number of the stacked MixSTEs: $L$, 2D pose sequence: $P_{N,T} = \{p_0, ... p_N, T\}$, Dimension of attention mechanism: $d$.

Output: High-dimensional output $F_l$ for each sequence

$T, N \leftarrow$ shape of $P_{N,T}$
$F_{N,T} = \text{LinearProjection}(P_{N,T})$

for $l \leftarrow 0$ to $L - 1$

if $l = 0$ then

$F_{N,T}^l \leftarrow \text{Spatio-Temporal Position } E_{pos}$

// Spatial Block
$S_{0:N}^l \leftarrow \text{Exact } N \text{ dimension of } F_{N,T}$
$AS = \text{Spatial Attention of } \{p_0, ... p_N\}$
$S_{0:N}^l = S_{0:n}^l + AS$
$S_{0:N}^l = S_{0:n}^l + MLP(S_{0:N}^l)$
$F_{N,T}^l \leftarrow S_{0:N}^l$

// Temporal Block
$T_{0:T}^l \leftarrow \text{Cross } N \text{ and } T \text{ Dimension of } F_{N,T}^l$
$AT = \text{Temporal Attention} \text{ for each joint } \{p_0, ... p_T\}$
$T_{0:T}^l = \text{Concat}(\{AT_0, ... AT_T\})$
$T_{0:T}^l = T_{0:T}^l + AT$
$T_{0:T}^l = T_{0:T}^l + MLP(T_{0:T}^l)$
$F_{N,0:T}^l \leftarrow T_{0:T}^l$

return $F_{N,0:T}^{L-1}$
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:

$$MPE(p_i, gt_i) = \frac{1}{T} \sum_{j=1}^{T} \| p_{j,i} - gt_{j,i} \|_2^2,$$  \hspace{1cm} (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 [3] 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) $L_t$ is defined as:

$$L_t = \frac{1}{NT} \sum_{j=2}^{T} \sum_{i=1}^{N} \| (p_{j,i} - p_{j-1,i}) \|_2^2,$$  \hspace{1cm} (3)

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

The MPJVE $L_m$ is also utilized in our model to improve the motion coherence [5] between the predicted poses and ground truth.

During the training stage, $\lambda_t$ and $\lambda_m$ are applied to weight $L_t$ and $L_m$. Therefore we train the network in an end-to-end manner with the multi loss function:

$$L = L_w + \lambda_t L_t + \lambda_m L_m.$$  \hspace{1cm} (4)

C. Additional Results

Comparison with PoseFormer. We compare the parameters, memory occupy, and training time per epoch of our model with PoseFormer [10]. 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 [5, 10].

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, which demonstrates that the strategy keeps the performance and successfully reduces the training time.
Table 8. Comparison with PoseFormer [10] in terms of frame per second (FPS), computing cost for each frame (FLOPs), and MPJPE. The evaluation is performed on Human3.6M testset S9, S11 under Protocol 1 with CPN [1] as the 2D pose detector. Computation is done on a single GTX 2080Ti GPU.

<table>
<thead>
<tr>
<th>Methods</th>
<th>FPS↑</th>
<th>FLOPs (M)↓</th>
<th>MPJPE↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>PoseFormer [10] (T=81)</td>
<td>288</td>
<td>1593</td>
<td>44.3</td>
</tr>
<tr>
<td>Ours (T=81)</td>
<td>965</td>
<td>965</td>
<td>42.5</td>
</tr>
<tr>
<td>Ours (T=243)</td>
<td>897</td>
<td>645</td>
<td>40.9</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Input Length</th>
<th>Sample Strategy</th>
<th>MPJPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>Ours (interval=27)</td>
<td>54.3</td>
</tr>
<tr>
<td>27</td>
<td>interval=9</td>
<td>56.9</td>
</tr>
<tr>
<td>27</td>
<td>interval=3</td>
<td>67.3</td>
</tr>
<tr>
<td>27</td>
<td>interval=1</td>
<td>78.8</td>
</tr>
</tbody>
</table>

Discussion of Sparse Attention. To further explore the sparse attention for our proposed method, we experiment some recent sparse attention works [2, 6, 9, 11]. 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 [8] 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 evaluate the proposed model on the Human3.6M dataset test set S11 with SittingDown action. The spatial attention 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 [1] 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.

References

[5] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with tem-


Figure 10. Qualitative Results of all heads attention maps among body joints. The x-axis (horizontal) and y-axis (vertical) correspond 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 (horizontal) 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 reconstructed 3D poses are shown.