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) [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 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 [4] 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 [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.

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 [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) $\mathcal{L}_t$ 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 [5] between the predicted poses and ground truth.

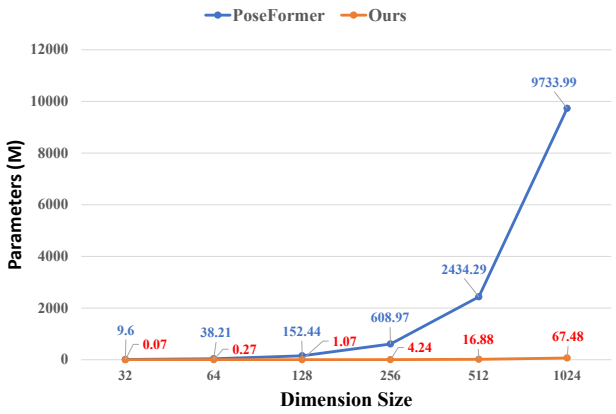
During the training stage, λ_t and λ_m are applied to weight $\mathcal{L}_t$ and $\mathcal{L}_m$. Therefore we train the network in an end-to-end manner with the multi loss function:

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

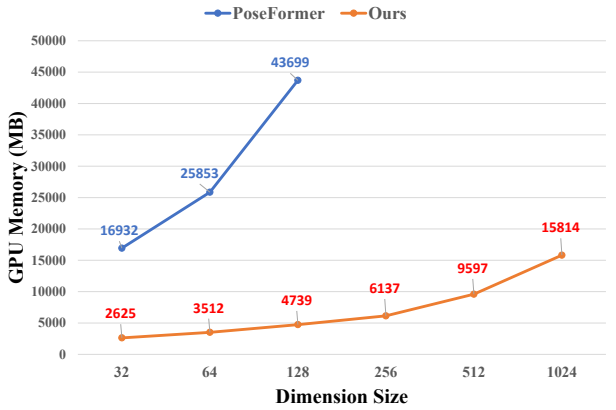
C. Additional Results

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

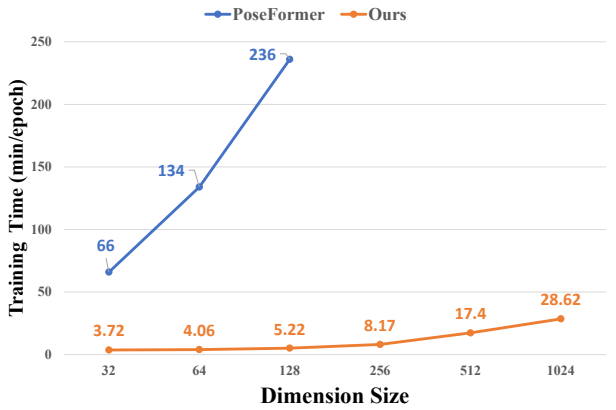
Effect of Data Sample Strategy. As shown in Figure 8, our stride data sample strategy results in fewer iterations to complete each training sample, thereby reducing overall training time. The stride data sample strategy is evaluated with different intervals. The max interval is equal to the input length, 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.



(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 [10]. The dimension size is the dimension of each query, key, and value in the encoders, which is the main factor of model size.

Methods	FPS↑	FLOPs (M)↓	MPJPE↓
PoseFormer [10] (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 [10] 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 [1] as the 2D pose detector. Computation is done on a single GTX 2080Ti GPU.

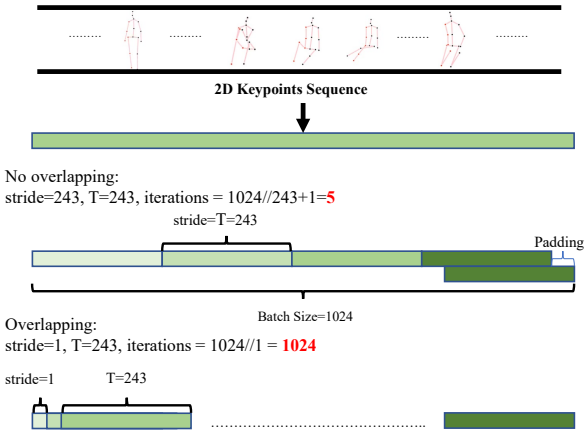


Figure 8. The processing example of stride data sample strategy. The stride example has fewer iterations than the example without stride sample, leading to less training time.

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.

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 evalauate the proposed model on the Human3.6M dataset

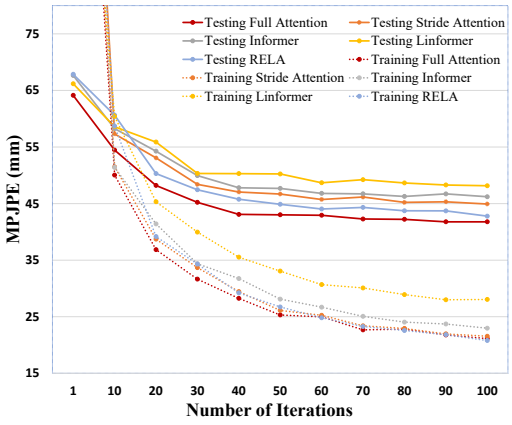


Figure 9. Comparison of different sparse attention and full attention mechanism for our method.

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

[1] 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. 1, 3

[2] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 3

[3] 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. 2

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. 1

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

poral convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019. 1, 2

[6] Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021. 3

[7] 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. 1

[8] 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. 3

[9] Biao Zhang, Ivan Titov, and Rico Sennrich. Sparse attention with linear units. *arXiv preprint arXiv:2104.07012*, 2021. 3

[10] 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

[11] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI*, 2021. 3

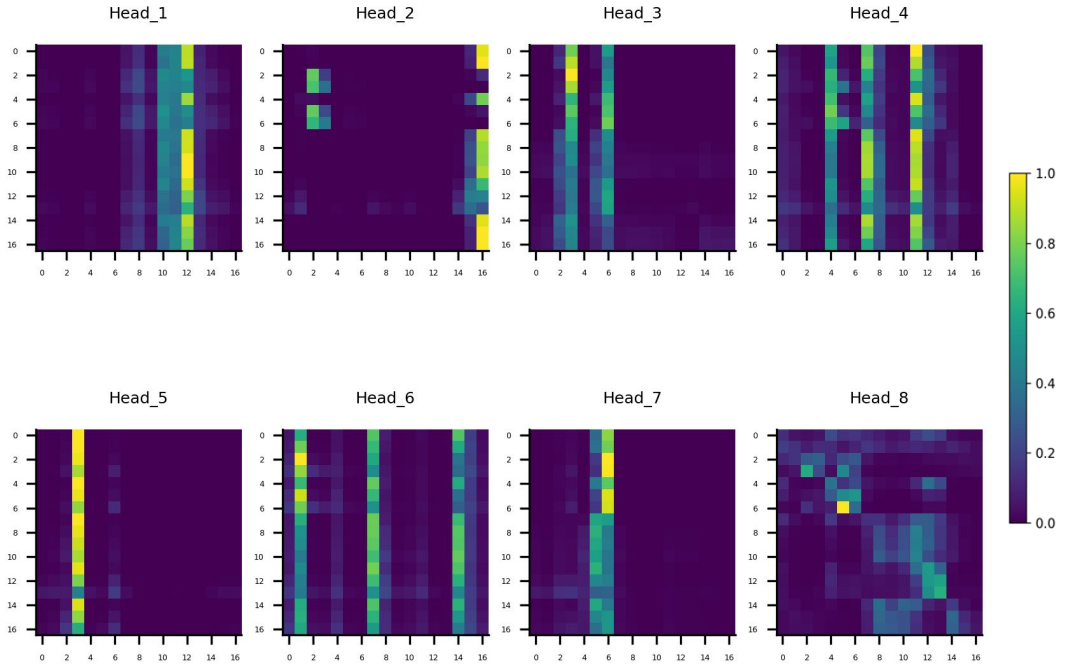


Figure 10. Qualitative Results of all heads attention maps among body joints. The x-axis (horizonal) and y-axis (vertical) to the joints queries and the predicted outputs, respectively. Each row shows the attention weight $w_{i,j}$ of the j-th query for the i-th output. The attention output is normalized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

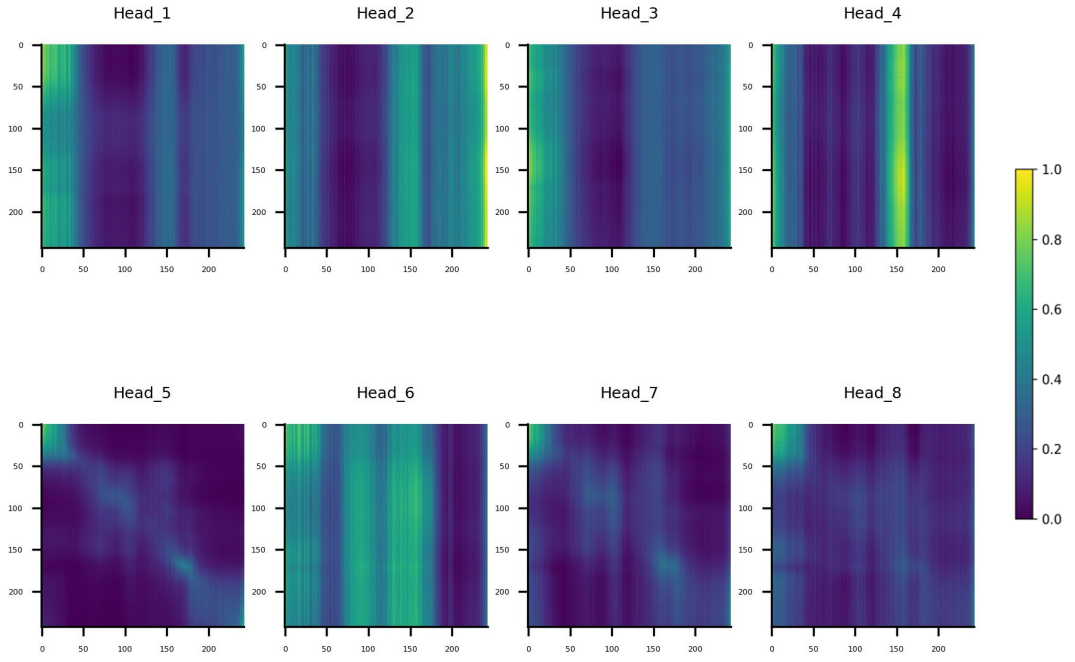


Figure 11. Qualitative Results of all heads attention maps among sequence frames. The x-axis (horizonal) and y-axis (vertical) correspond to the frames queries and the predicted outputs, respectively. Each row shows the attention weight $w_{i,j}$ of the j-th query for the i-th output. The attention output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