A Supplementary Material

A.1 Other Related Work

Hu *et al.* [5] also explores the spatial temporal masking strategy in the field of sign language recognition. In addition to being applied in different fields, our work differs from theirs in the following ways: 1) They set the coordinates of the masked joints to (0,0), which is a very common position (center of the image). Thus, this approach will lead to confusion between the masked and unmasked joints. In contrast, we replace the masked joints with learnable vectors. These special identifiers enable the model to distinguish whether a joint is masked or not. 2) They take all 2D poses as inputs, while we only take 2D unmasked poses as inputs, which greatly reduces the computational complexity of the encoder.

Liu *et al.* [10] learn 2D human pose embeddings for downstream tasks, which is similar to our work. Our method differs from theirs in these ways: 1) They aim to learn view-invariant embeddings by metric learning, while we aim to learn spatial temporal embeddings by solving the MPM task. 2) The masking strategy is utilized to increase the difficulty of pre-training in our method, while they use it to simulate the real-world occlusion situation. 3) We use Transformer, whose token design is more suitable for masking. 4) We explore the spatial temporal masking strategy, while they do not.

A.2 Detailed Network Architecture

The detailed architecture of STMO is illustrated in Fig. 1. For $N = 243$, we choose 485 frames and downsample them using TDS with the temporal downsampling rate s set to 2. The tensor sizes are shown in parentheses. For example, (243,34) denotes 243 frames and 34 channels, meaning that there are $J = 17$ joints in each frame, and each joint has 2 coordinates (x, y). $(243/3^i,256)$ means the temporal dimension is reduced by 3^i as 1D temporal convolution is used and the stride is the same as the kernel size $M = 3$. $i = 1, 2, ..., L_2$, where $L_2 = 5$ is the number of layers of MOFA.

A.3 Implementation Details

All experiments are carried out on a single GeForce GTX 3080Ti GPU. The proposed method is implemented using PyTorch [11]. For both stages, we train

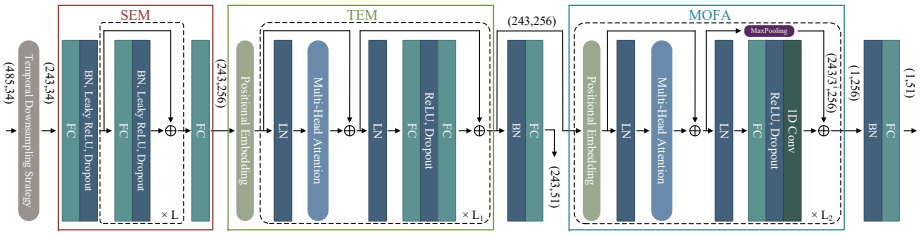


Fig. 1. Detailed architecture of the proposed STMO model in Stage II.

Table 1. Results on Human3.6M in millimeter under P-MPJPE. 2D poses detected by CPN are used as inputs. N is the number of input frames. The best result is shown in bold, and the second-best result is underlined.

P-MPJPE (CPN)	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Lin *et al.* [8] BMVC'19 (N=50)	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.* [12] CVPR'19 (N=243)	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
Xu *et al.* [16] CVPR'20 (N=9)	**31.0**	**34.8**	34.7	34.4	36.2	43.9	31.6	33.5	**42.3**	49.0	37.1	33.0	39.1	26.9	31.9	36.2
Liu *et al.* [9] CVPR'20 (N=243)	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
Wang *et al.* [15] ECCV'20 (N=96)	32.9	35.2	35.6	34.4	36.4	42.7	**31.2**	32.5	45.6	50.2	37.3	32.8	36.3	26.0	**23.9**	35.5
Chen *et al.* [1] TCSVT'21 (N=243)	33.1	35.3	33.4	35.9	36.1	41.7	32.8	33.3	42.6	49.4	37.0	32.7	36.5	25.5	27.9	35.6
Shan *et al.* [13] MM'21 (N=243)	32.5	36.2	33.2	35.3	35.6	42.1	32.6	31.9	42.6	**47.9**	36.6	**32.1**	34.8	24.2	25.8	35.0
Zheng *et al.* [18] ICCV'21 (N=81)	32.5	**34.8**	**32.6**	34.6	**35.3**	39.5	32.1	32.0	42.8	48.5	**34.8**	32.4	35.3	24.5	26.0	34.6
P-STMO (N=243)	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**

Table 2. Results on Human3.6M in millimeter under MPJPE. The ground truth of 2D poses are used as inputs. N is the number of input frames. The best result is shown in bold, and the second-best result is underlined.

MPJPE (GT)	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Pavllo *et al.* [12] CVPR'19 (N=243)	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
Lin *et al.* [8] BMVC'19 (N=50)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	32.8
Liu *et al.* [9] CVPR'20 (N=243)	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
Zeng *et al.* [17] ECCV'20 (N=243)	34.8	32.1	**28.5**	30.7	31.4	36.9	35.6	30.5	38.9	40.5	32.5	31.0	29.9	22.5	24.5	32.0
Chen *et al.* [1] TCSVT'21 (N=243)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	32.3
Zheng *et al.* [18] ICCV'21 (N=81)	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
Shan *et al.* [13] MM'21 (N=243)	29.5	30.8	28.8	29.1	30.7	35.2	31.7	**27.8**	**34.5**	**36.0**	30.3	**29.4**	28.9	24.1	24.7	30.1
P-STMO (N=243)	**28.5**	**30.1**	28.6	**27.9**	**29.8**	**33.2**	**31.3**	**27.8**	36.0	37.4	**29.7**	29.5	**28.1**	**21.0**	**21.0**	**29.3**

our model using the Adam [6] optimizer for 80 epochs. The initial learning rate is $1e^{-4}$ for Stage I and $7e^{-4}$ for Stage II. The learning rate decays by 3% after each epoch. The batch size is set to 160. We use horizontal flipping as the data augmentation approach during training and testing. The balance factor λ in the loss function is set to 1 empirically.

For SEM, we utilize an MLP block as the backbone. The number of sub-blocks is $L = 1$. The dimension of the latent features is set to 256. For TEM as well as the decoder in Stage I, we utilize a vanilla Transformer [14] as the backbone. The depth of Transformer for these two modules is set to 3,2 (for P-STMO-S) or 4,3 (for P-STMO), respectively. Besides, the dimension of the QKV matrices is 256. The number of heads in self-attention is 8. For MOFA, STE [7] is used as the backbone. The basic settings are the same as TEM. For $N = 27, 81, 243$, the depth of Transformer is set to 3, 4, 5, respectively. The stride of 1D convolution is set to 3 for each Transformer layer.

A.4 Additional Quantitative Results

We provide additional quantitative results of the proposed method on Human3.6M dataset. When CPN [2] is used as the 2D keypoint detector, our method achieves promising results under P-MPJPE (34.4mm), as shown in Table 1. To explore the lower bound of the proposed method, we utilize the ground truth of 2D poses as inputs. In this way, the input noise is removed. As shown in Table 2, P-STMO outperforms other methods with 29.3mm under MPJPE.

Table 3. Analysis on model hyperparameters. Only the number of parameters and FLOPS of Stage II are included in the calculation results. L_1, L_D are the depth of TEM and the decoder in Stage I respectively. d is the dimension of latent features of all Transformers.

L_1	L_D	d	Params(M)	FLOPs(M)	MPJPE↓
4	2	256	6.7	868.5	43.0
4	3	256	6.7	868.5	**42.8**
4	4	256	6.7	868.5	42.9
4	5	256	6.7	868.5	43.3
2	3	256	5.7	613.7	44.4
3	3	256	6.2	741.4	43.1
4	3	256	6.7	868.5	**42.8**
5	3	256	7.3	995.9	43.4
4	3	64	1.1	121.8	44.9
4	3	128	2.6	307.0	43.7
4	3	256	6.7	868.5	**42.8**
4	3	512	19.5	2755.1	45.5

A.5 Analysis on Model Hyperparameters

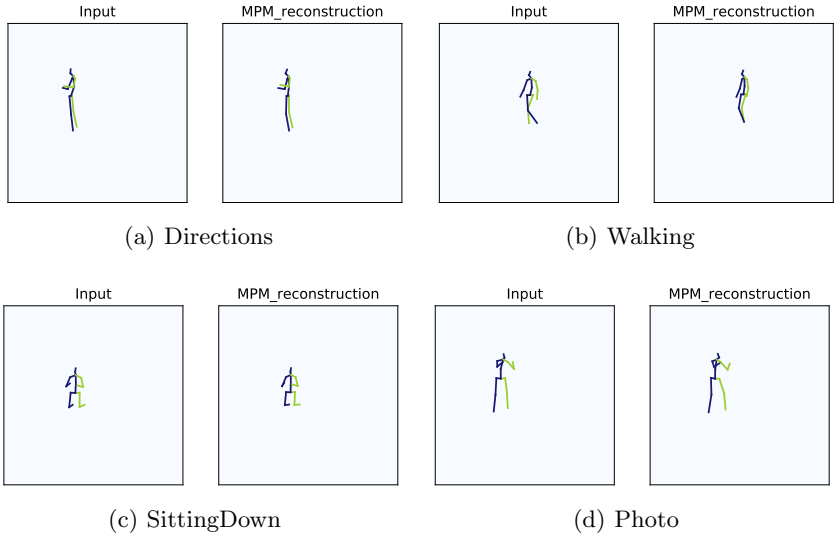
As shown in Table 3, we mainly investigate three hyperparameters in our method: the depth of TEM (L_1), the depth of the decoder in Stage I (L_D), and the dimension of latent features of all Transformers (d). Following [4], we adopt an asymmetric encoder-decoder design. Unlike classical auto-encoders, the depth of the decoder can be different from that of the encoder. Since we only care about the speed of inference, the number of parameters and FLOPs of the pre-training stage are not included in the calculation results. Thus, these results are not affected by different settings of L_D. The best performance is achieved when $L_1 = 4, L_D = 3, d = 256$.

A.6 Reconstruction Results in the Pre-Training Stage

In Fig. 2, we give some qualitative results of the 2D pose reconstruction in the pre-training stage. The 2D input sequence is masked with $q^{\mathrm{T}} = 0.8, m^{\mathrm{S}} = 2$, which means only 18.7% of the joints are visible to the network. It can be seen that the network is able to perform a rough recovery of the 2D poses in the original sequence by virtue of only a small number of visible joints.

A.7 Visualization of Multi-Head Self-Attention

We perform a subjective analysis of the self-attention mechanism in Transformer in both stages. The results are shown in Fig. 3-6. The x-axis (horizontal) and y-axis (vertical) correspond to the index of key (K) and query (Q) in the multi-head self-attention respectively. The output is the normalized attention weight. We set the input frame number to $N = 243$.



(a) Directions (b) Walking

(c) SittingDown (d) Photo

Fig. 2. Qualitative results of the 2D pose reconstruction in Stage I.

Stage I. We set the temporal masking ratio to $q^{\mathrm{T}} = 0.8$ and spatial masking number to $m^{\mathrm{S}} = 2$. Therefore, the number of unmasked frames is $a = (1 - q^{\mathrm{T}}) \cdot N = \lfloor 48.6 \rfloor = 48$. The number of masked frames is $b = N - a = 195$. The unmasked frames are fed to the encoder (SEM+TEM). The attention maps of TEM are shown in Fig. 3. Since the input is randomly masked, two adjacent frames in the unmasked frame sequence are most likely to be non-adjacent in the original sequence. Our goal is to recover all the masked frames between two adjacent frames in the unmasked frame sequence. Therefore, most heads of TEM focus on aggregating neighbouring information around the query frame. Others (e.g., head 6) capture long-term dependencies.

The attention maps of the decoder are shown in Fig. 4. An obvious cross can be seen in each attention map because we utilize the same implementation as [4]. Specifically, to improve the efficiency, we append a list of temporal padding embeddings after the encoded unmasked embeddings at the decoder side. Then, we add positional embeddings to these padding embeddings to restore their original positions. Therefore, the first half of the sequence (before frame 48) behaves differently from the second half (after frame 48). Each attention map can be divided into four matrices as follows. 1) The matrix $D_1 \in \mathbb{R}^{a \times a}$ in the upper left corner measures the self-attention of unmasked frames. The patterns are very similar to those in Fig. 3. 2) The matrix $D_2 \in \mathbb{R}^{a \times b}$ in the upper right corner measures the degree of attention paid to the masked frames by the unmasked frames. The values of most heads are 0, which means that the masked frames provide little help to the unmasked ones. A small number of heads behave differently (e.g., head 0), which can be explained by the need to interpret the continuity of the entire pose sequence. 3) The matrix $D_3 \in \mathbb{R}^{b \times a}$ in the bottom

Table 4. Analysis on the robustness in terms of MPJPE↓. ‡: w/o pre-training. RS: random shuffle.

Method	GT	GT+$\mathcal{N}$(0,0.01)	GT+$\mathcal{N}$(0,0.03)	GT+$\mathcal{N}$(0,0.05)	GT+RS	CPN
STMO-S‡	30.3	34.8	48.2	70.9	59.4	43.9
P-STMO-S	**29.3**(↓3.3%)	**33.7**(↓3.2%)	**46.9**(↓2.7%)	**69.5**(↓2.0%)	**57.7**(↓2.9%)	**43.0**(↓2.1%)

left corner measures the degree of attention paid to the unmasked frames by the masked frames. For a particular masked frame, it resorts to the unmasked frames that are close to it. As a result, some heads (e.g., head 4) exhibit a locally relevant pattern. In addition, non-local patterns can also be observed in other heads (e.g., head 3). 4) The matrix $D_4 \in \mathbb{R}^{b\times b}$ in the bottom right corner measures the self-attention of masked frames. Since the masked frames mainly obtain information from the unmasked frames, the self-attention pattern of masked frames is not pronounced. Nevertheless, we can still find some patterns learned by a small number of heads, such as capturing long-term dependencies (e.g., head 0) and short-term dependencies (e.g., head 1).

Stage II. After the pre-trained model is loaded into STMO, TEM is tuned to acquire long- and short-term information that contributes to the overall 3D pose sequence prediction. As illustrated in Fig. 5, Transformer learns a variety of patterns. The local patterns (e.g., head 0) focus on a small region around the query frame, while the global patterns (e.g., head 5) exploit relevant features over a longer time span.

As we use STE [7], a Transformer-based method, as the backbone network of MOFA, its attention maps can also be visualized in Fig. 6. For $N = 243$, the depth of Transformer is 5. Only the first 4 layers are shown. STE uses 1D convolution to reduce the temporal dimension layer by layer. As we set the stride to 3, the number of frames in each layer is reduced to 1/3 of that in the previous layer. A vertical line can be seen in all heads because MOFA is designed to predict the 3D pose in the current frame. Transformer forces all frames to attend to the current frame and its neighbours. Besides, the multi-head attention maps of MOFA and TEM show different patterns. The divergence of the functionality of these two modules is attributed to the presence of multi-frame loss.

A.8 Qualitative Results on in-the-wild Videos

We train our method on Human3.6M dataset and evaluate on in-the-wild videos. We use AlphaPose [3] as the 2D keypoint detector to generate 2D poses. As shown in Fig. 7, our method generalizes well to in-the-wild videos that often contain rare or unseen poses in the training set.

A.9 Generalization Ability and Robustness

To validate the generalization ability of the proposed pre-training method, we conduct experiments on Human3.6M, where three camera views (cam 0,1,2) are

used in Stage I and the other camera view (cam 3) is used in Stage II. The results show that P-STMO-S (w/ pre-training) and STMO-S (w/o pre-training) achieve 44.4mm and 45.2mm MPJPE respectively, which demonstrates that pre-training on several cameras can help the network generalize to a different camera view.

To validate the robustness of the proposed pre-training method, we utilize the ground truth (GT) of 2D keypoints as noiseless inputs, to which Gaussian noise with different σ is added. Note that CPN can be regarded as adding complex noise introduced by the 2D detector to the ground truth. Besides, the random shuffle (RS) strategy is tested. This strategy randomly disrupts the order of the input frames while leaving the order of the output frames unchanged. Table 4 shows pre-training still delivers performance gains in the case of input noise and RS, which verifies its robustness.

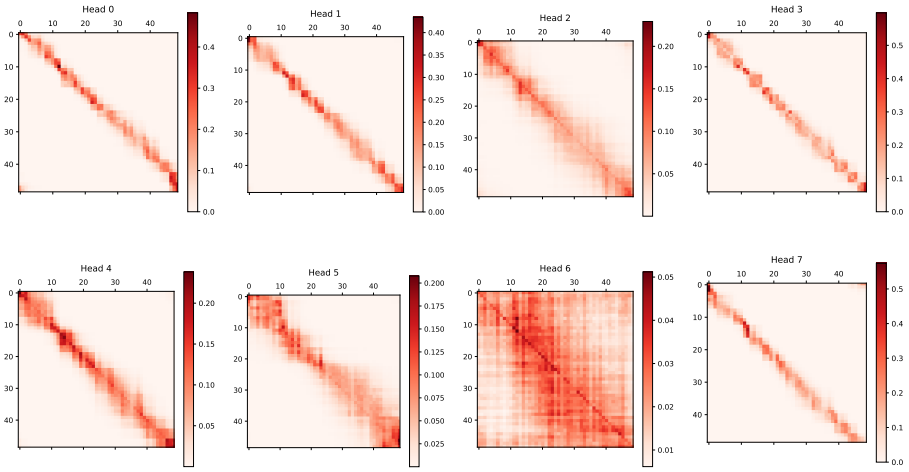


Fig. 3. Visualization of multi-head attention maps of TEM in Stage I.

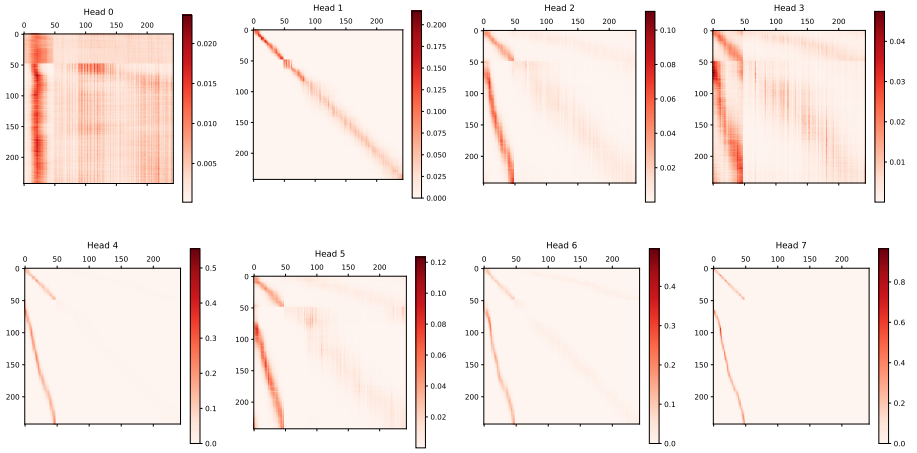


Fig. 4. Visualization of multi-head attention maps of the decoder in Stage I.

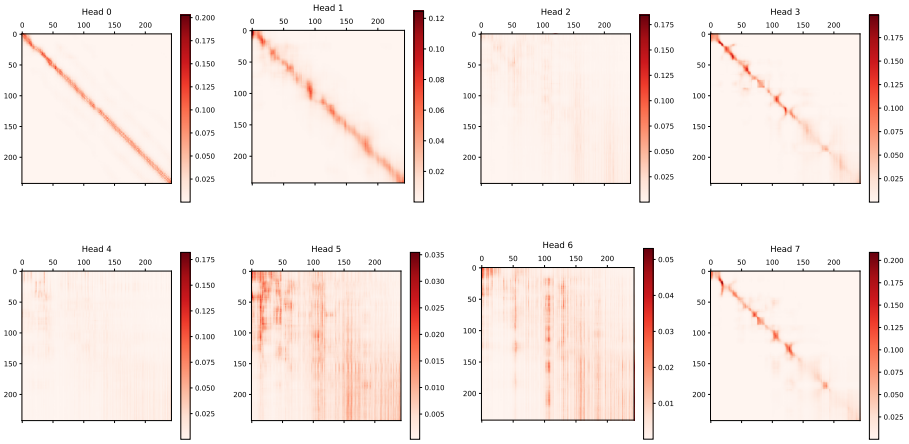
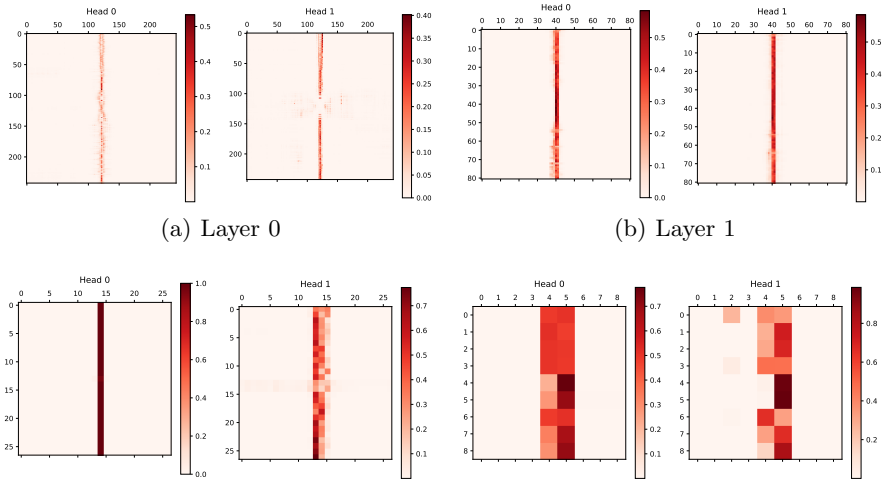


Fig. 5. Visualization of multi-head attention maps of TEM in Stage II.



(a) Layer 0 (b) Layer 1

(c) Layer 2 (d) Layer 3

Fig. 6. Visualization of multi-head attention maps of MOFA in Stage II. We only show the first two heads in each Transformer layer.

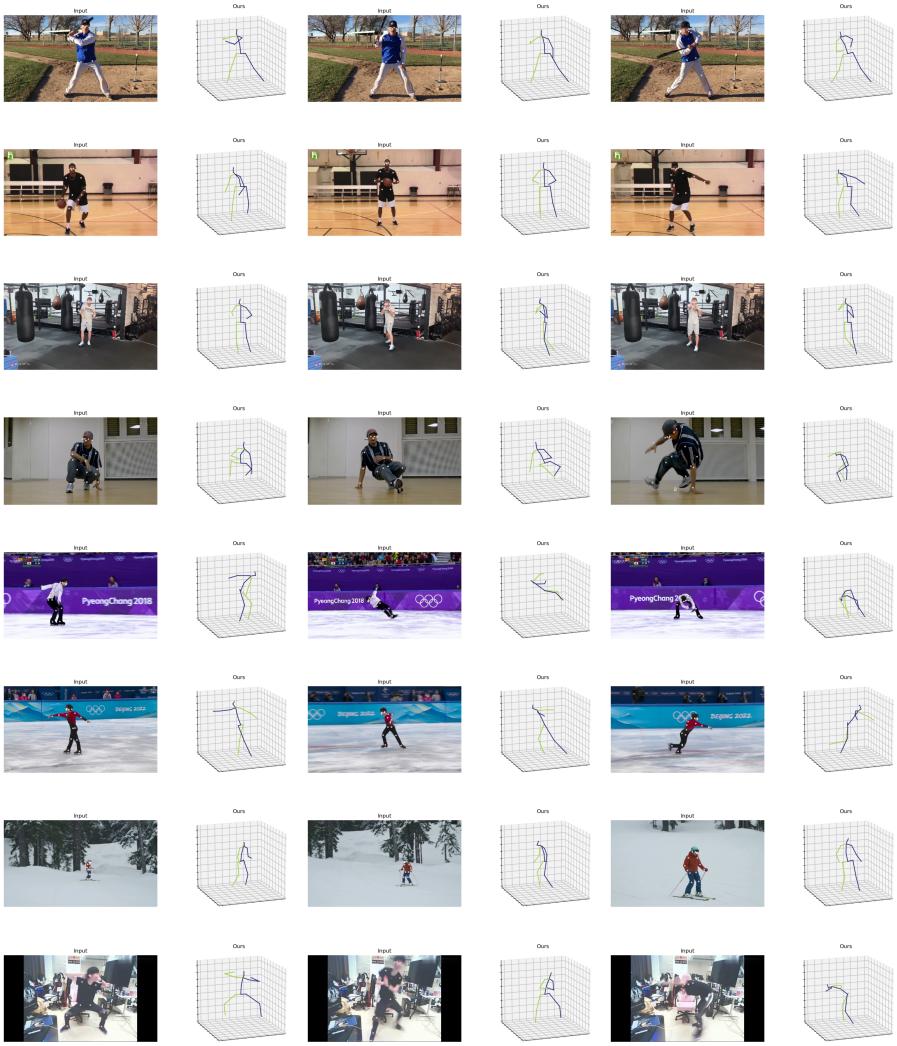


Fig. 7. Qualitative results of the proposed method on in-the-wild videos.

References

1. Chen, T., Fang, C., Shen, X., Zhu, Y., Chen, Z., Luo, J.: Anatomy-aware 3d human pose estimation with bone-based pose decomposition. IEEE Transactions on Circuits and Systems for Video Technology (2021)
2. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7103–7112 (2018)
3. Fang, H.S., Xie, S., Tai, Y.W., Lu, C.: Rmpe: Regional multi-person pose estimation. In: Proceedings of the IEEE international conference on computer vision. pp. 2334–2343 (2017)
4. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16000–16009 (2022)
5. Hu, H., Zhao, W., Zhou, W., Wang, Y., Li, H.: Signbert: Pre-training of hand-model-aware representation for sign language recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11087–11096 (2021)
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2015)
7. Li, W., Liu, H., Ding, R., Liu, M., Wang, P., Yang, W.: Exploiting temporal contexts with strided transformer for 3d human pose estimation. IEEE Transactions on Multimedia (2022)
8. Lin, J., Lee, G.H.: Trajectory space factorization for deep video-based 3d human pose estimation. In: Proceedings of the British Machine Vision Conference (BMVC) (2019)
9. Liu, R., Shen, J., Wang, H., Chen, C., Cheung, S.c., Asari, V.: Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5064–5073 (2020)
10. Liu, T., Sun, J.J., Zhao, L., Zhao, J., Yuan, L., Wang, Y., Chen, L.C., Schroff, F., Adam, H.: View-invariant, occlusion-robust probabilistic embedding for human pose. International Journal of Computer Vision **130**(1), 111–135 (2022)
11. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32**, 8026–8037 (2019)
12. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 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. pp. 7753–7762 (2019)
13. Shan, W., Lu, H., Wang, S., Zhang, X., Gao, W.: Improving robustness and accuracy via relative information encoding in 3d human pose estimation. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 3446–3454 (2021)
14. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
15. Wang, J., Yan, S., Xiong, Y., Lin, D.: Motion guided 3d pose estimation from videos. In: European Conference on Computer Vision. pp. 764–780. Springer (2020)

16. Xu, J., Yu, Z., Ni, B., Yang, J., Yang, X., Zhang, W.: Deep kinematics analysis for monocular 3d human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 899–908 (2020)
17. Zeng, A., Sun, X., Huang, F., Liu, M., Xu, Q., Lin, S.: Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. In: European Conference on Computer Vision. pp. 507–523. Springer (2020)
18. Zheng, C., Zhu, S., Mendieta, M., Yang, T., Chen, C., Ding, Z.: 3d human pose estimation with spatial and temporal transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11656–11665 (2021)