A Appendix

A.1 Parallel Representation of NC-RetNet

In this section, we formulate the parallel representation of NC-RetNet, especially the D matrix. Denote the length of training sequences as L and the training chunk size as T, then the D matrix is as in Eq. (6).

$$D_{nm} = \begin{cases} \gamma^{|n-m|}, & m \leq \lceil n/T \rceil * T \\ 0, & m > \lceil n/T \rceil * T \end{cases}, n, m \in \{1, ..., L\} \tag{6}$$

Given this D matrix, the parallel representation of NC-RetNet is the same as Eq. (1). This representation is equivalent to the chunkwise recurrent representation using the D matrix in Eq. (4), because only the temporal information from previous chunks ($m \leq \lfloor n/T \rfloor * T$) and the current chunk ($\lfloor n/T \rfloor * T < m \leq \lceil n/T \rceil * T$) is available in both representations. Therefore, the NC-RetNet trained in the parallel representation can be used to infer test sequences directly in the chunkwise recurrent representation.

A.2 Overall Architecture

In this section, we introduce the model architecture built on MixSTE. Given a stream of 2D keypoints, the model processes the stream every T frames, where

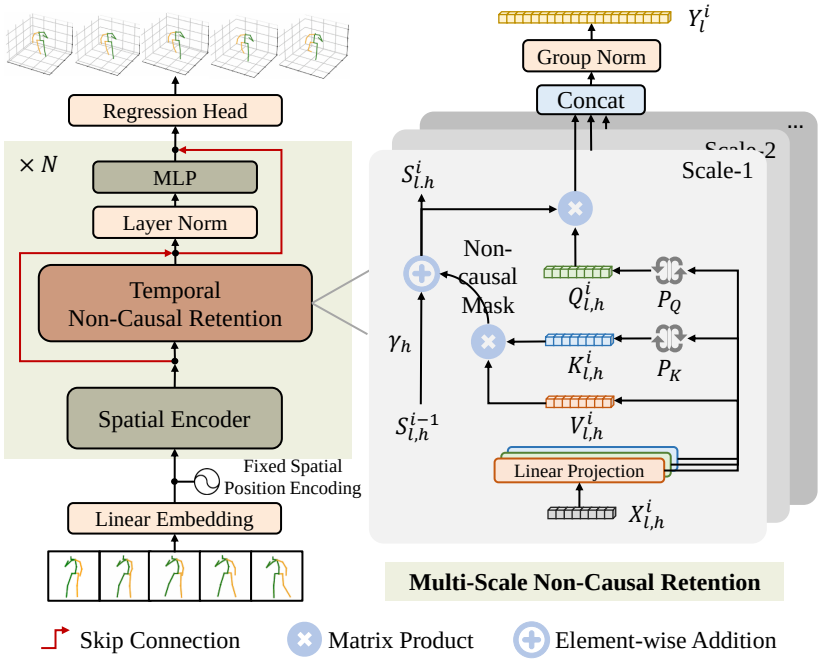

Fig. 5: Architecture of the human pose estimation model based on NC-RetNet.

the T frames is referred to as a *chunk*. The 2D keypoints in the i^{th} chunk are first embedded using a linear layer, and then the fixed spatial position encoding is added. The resulting embedding is then passed into the spatial-temporal encoder. There are N blocks in the spatial-temporal encoder, and each block consists of a transformer-based spatial encoder and an NC-RetNet based temporal encoder. This alternating way of integrating spatial and temporal information is borrowed from [13]. Since the transformer-based spatial encoder has been widely used by previous methods, we only formulate the temporal encoder below.

The temporal encoder is composed of a Multi-Scale Non-Causal Retention Module, a Layer Normalization [1] and an MLP. Similar with the transformer block, the skip connection is applied. Denote the input tokens of the i^{th} to the l^{th} temporal encoder as $X_l^i \in \mathbb{R}^{T\times J\times d}$, where J is the number of human joints. The Multi-Scale Non-Causal Retention Module divides the input tokens into H parts along the feature dimension, i.e. $X_{l,h}^i \in \mathbb{R}^{T\times J\times (d/H)}, h \in \{1, ..., H\}$. Each part is then processed by a simple Non-Causal Retention Module as described in Sec. 3.2, and the output tokens of H simple modules are concatenated together before being fed into a Group Normalization [11]. The γ_h for each simple module is different, so the whole module is termed Multi-Scale Non-Causal Retention (MS-NC-Retention), which can be formulated in Eq. (7). Since different decay coefficients are integrated into one MS-NC-Retention, it can capture the temporal information of multiple frequencies.

$$\begin{aligned} Y_{l,h}^i, S_{l,h}^i &= \text{NC-Retention}(X_{l,h}^i, S_{l,h}^{i-1}) \\ Y_l^i &= \text{GroupNorm}(\text{Concat}(Y_{l,h}^i)), h \in \{1, ..., H\} \end{aligned} \tag{7}$$

Note that we does not formulate the joint-related decay coefficients above for simplicity. Intuitively speaking, this design further allows capturing temporal information of different ranges of frequencies for different joints.

Finally, a regression head built on MLP is used to regress the 3D poses within the current chunk.

A.3 Hyper-parameter Settings

We use the Adam optimizer [4] and train the model for 150 epochs. The dimension of the model d is 512, and the number of spatial-temporal blocks N is 8. We use DropPath [5] when training and the drop path rate is 0.1. Data augmentation of horizontal flipping is applied during training and testing. To diversify the chunk segmentation, a random shift is implemented on both datasets. Specifically, the initial index for segmenting the sequences is not consistently the first frame, but a randomized one.

For training on the Human3.6M dataset, we set L to 900 and sample the training sequences with a stride of 450. The initial learning rate is 4e-5 and the learning rate decay for each epoch is 0.99. The batch size on each GPU is 1.

For training on the MPI-INF-3DHP dataset, we set L to 600 and sample the training sequences with a stride of 600. The initial learning rate is 1e-4, and the learning rate decay for each epoch is 0.98. The batch size on each GPU is 2.

A.4 Comparison with STCFormer

Although the reported accuracy of our method at $T = 243$ does not surpass STCFormer [10] with a clear margin, we argue that our method is better. First, we does not use the tricks used by STCFormer, including Temporal Downsampling Strategy [8] and integrating results with $T = 81$. Without the latter one, STCFormer only achieves an MPJPE of 41.0 mm when $T = 243$. Second, the STCFormer adopts the seq2frame framework, which only predicts one frame given a lot of input frames. In contrast, our method predicts the results for all input frames, which is a much more efficient way.

A.5 Comparison with More SOTA Methods

In this section, we list the results of more SOTA methods in Tab. 6 (top 5 rows). It can be seen that our method is still the best. We also conduct the experiment by building our model on another state-of-the-art architecture, DSTformer proposed by MotionBERT [15]. The results are shown in Tab. 6 (bottom 4 rows). In this experiment, the model estimates the image-normalized 3D pose and uses additional scaling factors during testing as MotionBERT does. The training chunk size is 243. With the same architecture and data, our method is better than the original DSTformer when T is 243, and the performance of our method does not deteriorate rapidly as T becomes smaller. These results demonstrate the effectiveness of our module with different architectures.

Table 6: Comparison of MPJPE on the Human3.6M dataset with more SOTA methods. SH is short for Stacked Hourglass [6]. T is the test chunk size.

	Publication	2D detector	T	MPJPE
PoseformerV2 [14]	CVPR'23	CPN	243	45.2
UPS [3]	CVPR'23	CPN	243	40.8
Einfalt *et al.* [2]	WACV'23	CPN	351	41.7
GLA-GCN [12]	ICCV'23	CPN	243	44.4
Ours + MixSTE		CPN	243	**40.4**
MotionBERT [15]	ICCV'23	SH	243	39.2
Ours + DSTformer		SH	27	40.4
Ours + DSTformer		SH	81	39.4
Ours + DSTformer		SH	243	**38.9**

A.6 Ablations on Joint-Related Decay Coefficients

To validate the effectiveness of the joint-related decay coefficients, we train the models using the same γ for all human joints, and we test four values, $1 - 1/4$, $1 - 1/8$, $1 - 1/12$ and $1 - 1/16$. The results are in Tab. 7. It can be seen that using none of the four base γ's for all human joints beats using joint-related decay coefficients, demonstrating the effectiveness of joint-related decay coefficients.

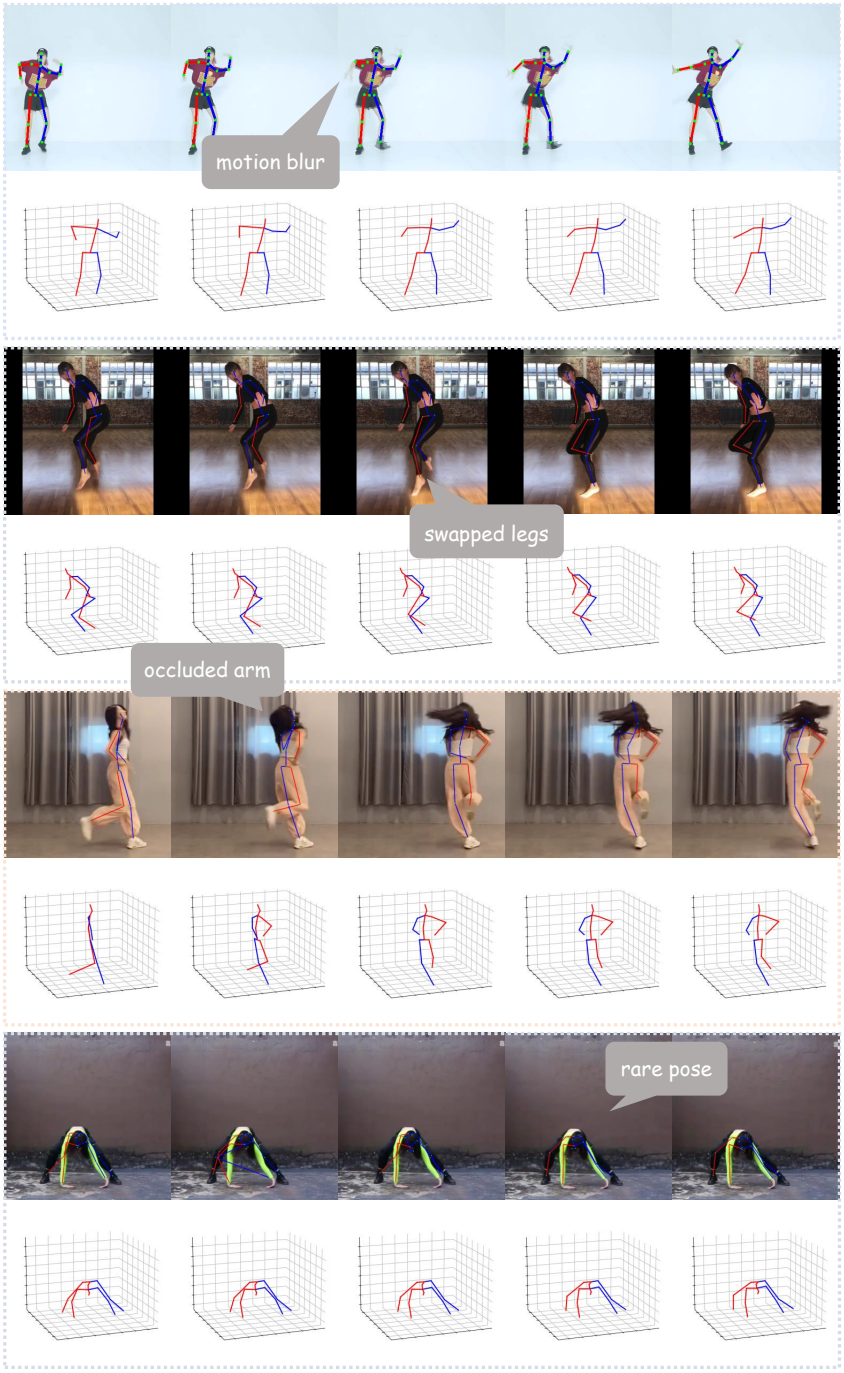

Fig. 6: Visualization of some in-the-wild examples predicted by our method.

Table 7: Ablation study of effect of joint-related decay coefficients compared with using the same decay coefficient for all human joints.

base γ	MPJPE	PA-MPJPE
$1-1/4$	41.0	33.0
$1-1/8$	40.7	32.7
$1-1/12$	40.8	32.8
$1-1/16$	41.1	33.1
Joint-Related	**40.4**	**32.5**

A.7 More Visualization Results

Performance on In-the-wild Data Using YOLOv3 [7] as the bounding box detector and HRNet [9] as the 2D keypoints detector, we test our method on some in-the-wild examples, and the results are shown in Fig. 6. It can be seen that our method is able to predict accurate and continuous 3D poses despite noisy 2D input caused by motion blur, occlusion or rare pose. This shows that our method generalizes well to unseen data samples. More results of in-the-wild videos are included in the ZIP file.

Visualization of Feature Similarity We visualize the similarity of joint features under different timesteps, seen in Fig. 7. The two skeletons at $t1$ and $t2$ are similar, and the difference mainly lies in the pose of upper arms. Therefore, the similarities of other human parts are very high. For example, the similarities between $t1$ and $t2$ of the left hip, right hip, head and nose reach nearly 1.0. But those of the left elbow, right elbow, left wrist and right wrist are lower (approximately 0.95). This result demonstrates that our model extracts meaningful features for human joints that can reflect the similarity. Moreover, it can

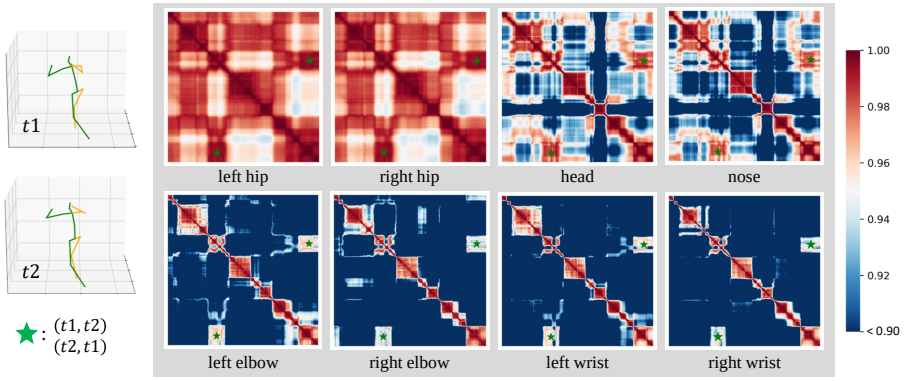

Fig. 7: Visualization of the similarity of the features along the time axis of different human joints.

be seen that the similarities along the time axis of the limb joints are generally lower than those of the torso and head joints. Specifically, most of the similarities of the limb joints are lower than 0.9. This results show that the design of the joint-related decay coefficients is reasonable.

A.8 Throughput

We test the throughput of the above pipeline, which includes YOLOv3, HRNet, and NC-RetNet on a single GeForce RTX 3090 GPU. The throughput of 2D bounding box detection and 2D keypoint detection is 16.23 items per second, while the throughput of 2D-to-3D lifting using our method is 1059.48 items per second. Overall, the throughput of this pipeline is 15.99 items per second, which is very close to the detection and 2D estimation step. Therefore, our 2D-to-3D lifting method is highly efficient.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016) 2
2. Einfalt, M., Ludwig, K., Lienhart, R.: Uplift and upsample: Efficient 3d human pose estimation with uplifting transformers. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2903–2913 (2023) 4, 11, 3
3. Foo, L.G., Li, T., Rahmani, H., Ke, Q., Liu, J.: Unified pose sequence modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13019–13030 (2023) 3
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 2
5. Larsson, G., Maire, M., Shakhnarovich, G.: Fractalnet: Ultra-deep neural networks without residuals. arXiv preprint arXiv:1605.07648 (2016) 2
6. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14. pp. 483–499. Springer (2016) 3
7. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018) 5
8. Shan, W., Liu, Z., Zhang, X., Wang, S., Ma, S., Gao, W.: P-stmo: Pre-trained spatial temporal many-to-one model for 3d human pose estimation. In: European Conference on Computer Vision. pp. 461–478. Springer (2022) 10, 11, 14, 3
9. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: CVPR (2019) 3, 5
10. Tang, Z., Qiu, Z., Hao, Y., Hong, R., Yao, T.: 3d human pose estimation with spatio-temporal criss-cross attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4790–4799 (2023) 9, 10, 11, 12, 13, 14, 3
11. Wu, Y., He, K.: Group normalization. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018) 2

12. Yu, B.X., Zhang, Z., Liu, Y., Zhong, S.h., Liu, Y., Chen, C.W.: Gla-gcn: Global-local adaptive graph convolutional network for 3d human pose estimation from monocular video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8818–8829 (2023) 3
13. Zhang, J., Tu, Z., Yang, J., Chen, Y., Yuan, J.: Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13232–13242 (2022) 2, 8, 9, 10, 11, 12, 13, 14
14. Zhao, Q., Zheng, C., Liu, M., Wang, P., Chen, C.: Poseformerv2: Exploring frequency domain for efficient and robust 3d human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8877–8886 (2023) 2, 3
15. Zhu, W., Ma, X., Liu, Z., Liu, L., Wu, W., Wang, Y.: Motionbert: A unified perspective on learning human motion representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15085–15099 (2023) 4, 9, 3