3D Human Pose Estimation via Non-Causal Retentive Networks

Kaili Zheng[1], Feixiang Lu[2], Yihao Lv[2], Liangjun Zhang[2], Chenyi Guo[1]✉, and Ji Wu[1,3,4]✉

[1] Department of Electronic Engineering, Tsinghua University
[2] Baidu Research
[3] College of AI, Tsinghua University
[4] Beijing National Research Center for Information Science and Technology
zkl22@mails.tsinghua.edu.cn

Abstract. Temporal dependencies are essential in 3D human pose estimation to mitigate depth ambiguity. Previous methods typically use a fixed-length sliding window to capture these dependencies. However, they treat past and future frames equally, ignoring the fact that relying on too many future frames increases the inference latency. In this paper, we present a 3D human pose estimation model based on Retentive Networks (RetNet) that incorporates temporal information by utilizing a large number of past frames and a few future frames. The Non-Causal RetNet (NC-RetNet) is designed to allow the originally causal RetNet to be aware of future information. Additionally, we propose a knowledge transfer strategy, i.e., training the model with a larger chunk size and using a smaller chunk size during inference, to reduce latency while maintaining comparable accuracy. Extensive experiments have been conducted on the Human3.6M and MPI-INF-3DHP datasets, and the results demonstrate that our method achieves state-of-the-art performance. Code and models are available at https://github.com/Kelly510/PoseRetNet.

Keywords: 3D Human Pose Estimation · Temporal Dependency · Retentive Networks

1 Introduction

Monocular 3D Human Pose Estimation (HPE) aims to reconstruct the 3D positions of human body joints based on monocular observations. This popular computer vision task has a wide range of applications, including action recognition [44], human-robot interaction [37] and motion analysis [11]. Most of the previous works [2,22,39–41,45,48] adopt the 2D-to-3D lifting pipeline which predicts 3D human pose based on 2D keypoint detection results. It is challenging due to the depth ambiguity issue, namely, one 2D detection result may correspond to multiple 3D human skeletons.

✉ denotes corresponding author.

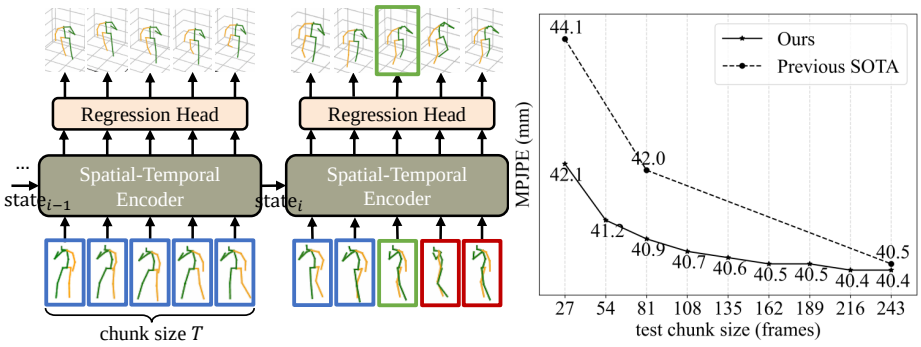

Fig. 1: (Left) The framework of our method, which utilizes long-term historical information from the cross-chunk state and relies on only a few future frames within the chunk. The past, current, and future frames are denoted by blue, green, and red borders, respectively. **(Right)** Comparison of Mean Per-Joint Position Error (MPJPE) on the Human3.6M dataset under different test chunk sizes. Our method outperforms previous state-of-the-art remarkably, especially under small chunk sizes.

To mitigate the depth ambiguity, monocular 3D human pose estimation models usually take multiple frames as the input and exploit additional temporal dependencies of human pose to reduce the ambiguity [1, 27, 29, 43, 46, 49, 50]. Specifically, a sliding-window of fixed length is usually adopted to capture the temporal dependencies, where the length of window is referred to as the *number of frames* or *chunk size*. A larger chunk size typically results in better accuracy performance as it allows for the perception of more long-range temporal information. However, previous methods treat past and future frames equally, and a larger chunk size also means that the model relies on the arrival of more future frames before inference, which significantly increases the inference latency. For instance, consider the seq2frame framework, which aims to predict the 3D pose of the center frame among the input frames. If the chunk size is 243 and the input frame rate is 10 Hz, the inference latency will be (243 1) 2 10 = 12:1 seconds. For seq2seq framework in the same case, the inference latency for the first frame within the chunk is (243 1) 10 = 24:2 seconds and that for the last frame is zero. The average latency is 12:1 seconds as well. This is considerably longer than the forward time of the model itself.

To address this problem, we propose a 3D human pose estimation model based on Retentive Networks (RetNet) [35]. Fig. 1(left) illustrates the framework of our method. Different from previous methods that use similar amounts of past and future frames to incorporate temporal information, our method mainly extracts temporal information from past frames (blue) and uses only a few future frames (red) within the current chunk for refinement. The RetNet can easily capture long-term historical information by using the cross-chunk state, and the Non-Causal RetNet (NC-RetNet) is further designed to make the originally causal RetNet be aware of the future frames. Moreover, we develop a knowl-

edge transfer strategy of training the model with a large chunk size and using a small chunk size during inference. Thanks to the long-term historical information brought by the cross-chunk state, decreasing the test chunk size does not significantly affect performance, as shown in Fig. 1(right), but greatly reduces the inference latency.

Extensive experiments have been conducted on two datasets, Human3.6M [14] and MPI-INF-3DHP [23], both quantitatively and qualitatively. The results demonstrate that our method outperforms state-of-the-art with a clear margin in terms of accuracy and continuity, especially when the model infers with a small chunk size. Our method even surpasses state-of-the-art with a smaller chunk size during inference. The ablation study also validates the efficacy of the components in our method. Our main contributions can be summarized as follows.

1. This is the first study to investigate the potential of RetNet in 3D human pose estimation. And we introduce NC-RetNet to extract temporal information, which leverages past frames through the cross-chunk state and a limited number of future frames within the chunk.
2. The NC-RetNet can be trained using a large chunk size and infer using a small chunk size without significant performance deterioration, but with a notable decrease in inference latency.
3. Extensive experiments have been conducted and the results demonstrate that our method is the state-of-the-art in terms of accuracy and continuity, especially when the test chunk size is small.

2 Related Work

2.1 3D Human Pose Estimation

Monocular 3D human pose estimation is a fundamental computer vision task with a broad range of applications. Direct estimation of the 3D positions of human joints from raw image pixels [26, 34] is difficult not only because of the complexity of extracting image features, but also due to the lack of image-3D data pairs. For these reasons, Martinez *et al.* [22] propose to estimate 3D human pose in a two-stage manner: detect 2D keypoints from images first and then lift 2D to 3D. Since this approach can utilize existing 2D pose estimation systems [3, 19, 25, 33, 42] and a large amount of 3D motion capture data, it has received a lot of attention. In this paper, we also focus on the 2D-to-3D lifting task. Although there are methods such as [48] propose to leverage visual cues only to mitigate depth ambiguity, these methods are unable to produce reconstructions with good continuity. Therefore, temporal dependencies are very crucial for monocular human pose estimation models.

2.2 Exploitation of Temporal Dependencies

Previous methods mostly adopt four architectures to exploit temporal dependencies: CNN, RNN, GCN [15] and transformer [21]. For example, to model

the temporal dependencies of human motion, Pavllo *et al.* [27] propose a temporal convolution model that utilizes dilated temporal convolutions to capture long-term information and model the temporal dependencies of human motion. The temporal receptive field depends on the dilation ratio and the number of layers. Similarly, Choi *et al.* [5] utilizes GRU [7] to extract features from the past frames and future frames within a fixed-length window respectively before integration. Cai *et al.* [1] exploit graph convolutions [15] to model the graph structure of different human joints. Along the time axis, this method treats the joints at different time steps as the graph nodes where any two consecutive joints are adjacent in the graph. Poseformer [50] proposed by Zheng *et al.* is the first work to introduce transformers to 3D human pose estimation task. This model incorporates the Spatial Transformer Module to encode the geometric structure of the human pose in a single frame into a token, and the Temporal Transformer Encoder to model temporal dependencies between frames. Since then, a lot of works [8, 10, 16–18, 51] have emerged to explore the potential of transformers in 3D human pose estimation.

Although these methods leverage different architectures to extract spatial-temporal information from 2D sequences, they share a common framework that employs a fixed number of frames to predict the result. Moreover, the chunk size has a significant impact on the accuracy, and a larger chunk size is usually beneficial for performance. However, previous works have not taken into account that a larger chunk size also significantly increases the inference delay. This motivates us to develop a method that balances the accuracy and inference latency better.

2.3 Real-time Human Pose Estimation

In addition to accuracy, low inference latency is also desired for human pose estimation models in many scenarios, and significant efforts have been devoted to reducing the inference latency. On one hand, since human pose estimation models typically use a backbone model to extract image features, general-purpose lightweight backbones [9, 13, 30, 47], can be used directly to replace the backbone in HPE models [6]. On the other hand, simplifying the pipeline can improve the model's efficiency. For example, Vnect [24] is proposed to combine the bounding box detection, 2D keypoint detection, and 2D-to-3D lifting into one model. However, existing methods only focus on decreasing the forward time of the HPE models, but do not consider the inference latency caused by large chunk sizes, as our method does.

2.4 Length Extrapolation

Models in natural language processing are expected to be generalizable across sequences of varying lengths, particularly to sequences longer than the training samples. This desired property is called length extrapolation. To achieve this, the use of relative position embedding, such as RoPE [32] and xPos [36], is necessary because it does not require the input sequences to be of fixed length.

Additionally, there are methods [4, 28] to improve the format of the attention module to achieve length extrapolation. For example, ALIBI [28] proposes to subtract the absolute temporal distance of two tokens from the attention score, which enhances the performance on extremely long sequences. Our knowledge transfer strategy is similar to length extrapolation, except that we concentrate on the model's transition from large chunks to smaller ones.

3 Method

3.1 Preliminary

RetNet [35] is a sequence modeling network that produces a contextualized feature sequence of length L given an input sequence $X \in \mathbb{R}^{L \quad d}$. The basic module of RetNet is retention, which has three mathematically equivalent representations: parallel, recurrent, and chunkwise recurrent. We present a detailed explanation of these representations below.

Parallel Given the input sequence X, the query Q and key K are derived by applying the linear projection and RoPE. $P_q; P_k$ are the rotary position embedding for the query and the key respectively. The value V is obtained by the linear projection only. $D \in \mathbb{R}^{L \quad L}$ is the combination of causal masking and exponential decay with respect to the relative distance. denotes the element-wise product. Since the value in the mask is non-zero only when the reference token (m^{th}) is earlier than the target token (n^{th}), RetNet is a fully causal model.

$$\begin{aligned} &Q = P_q(XW_Q); K = P_k(XW_K); V = XW_V \\ &D_{nm} = \begin{cases} {}^{n \quad m}; & n \quad m \\ 0; & n < m \end{cases} \\ &\text{Retention}(X) = (QK^T \quad D)V \end{aligned} \tag{1}$$

Recurrent $S_n \in \mathbb{R}^{d \quad d}$ represents the state of time step n, and $Q_n; K_n; V_n$ is the value of the same $Q; K; V$ in Eq. (1) at time step n. This representation also shows that RetNet is entirely causal, as the output for the n^{th} frame depends solely on the previous state $S_{n \quad 1}$ and the n^{th} input.

$$\begin{aligned} &S_n = \quad S_{n \quad 1} + K_n^T V_n \\ &\text{Retention}(X_n) = Q_n S_n; n = 1; \ldots; L \end{aligned} \tag{2}$$

Chunkwise Recurrent This representation is the hybrid form of the above two representations. Suppose the input sequence is segmented into chunks of length T. Denote $X_{iT:(i+1)T}$ as $X_{[i]}$, where $[i]$ indicates the i-th chunk. Within the chunk, the model follows the parallel representation and the cross-chunk information is passed following the recurrent representation. The D here is similar

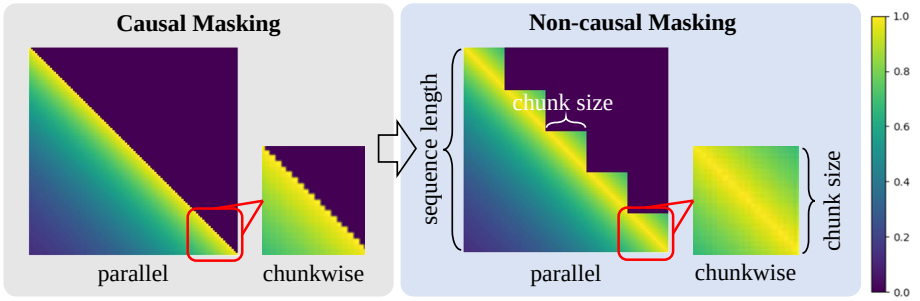

Fig. 2: (Left) The causal masks in the parallel and chunkwise recurrent representations of the original RetNet. The model can only perceive historical frames although there are several future frames in the current chunk. **(Right)** We propose Non-Causal RetNet (NC-RetNet), which utilizes all the frames within current chunk using the full mask and can be trained in parallel with the staircase-shaped mask.

to that in Eq. (1), but its shape changes from $L \quad L$ to $T \quad T$. The D's in Eq. (1) and Eq. (3) are illustrated in Fig. 2(left). and are both $T \quad d$ matrices and the r-th row of them is $^{T} \; ^{r} \; ^{1}$ and $^{r+1}$ respectively.

$$
\begin{aligned}
S_i &= K_{[i]}^{T}(V_{[i]} \qquad) + \quad {}^{T}S_{i \;\; 1} \\
\text{Retention}(X_{[i]}) &= \underbrace{(Q_{[i]}K_{[i]}^{T} \qquad D)V_{[i]}}_{\text{Inner-Chunk}} + \underbrace{(Q_{[i]}S_{i \;\; 1})}_{\text{Cross-Chunk}}
\end{aligned} \tag{3}
$$

Since low inference latency is required in real-time scenarios, the parallel representation is not suitable. Moreover, the recurrent representation is the special case of the chunkwise recurrent representation when $T = 1$. Therefore, we focus on the chunkwise recurrent representation of RetNet to design our human pose estimation model.

3.2 Non-Causal RetNet

Although the chunkwise recurrent representation of RetNet processes the input sequence chunk by chunk, it does not utilize all the information in the current chunk. As is shown in Fig. 2(left), the masking in the chunkwise recurrent representation is a lower triangular matrix. This means that the estimation of the current frame only uses the frames before it, regardless of the future frames within the chunk. However, leveraging certain future information can be very helpful for the accuracy of human pose estimation models.

To solve this problem, we modify the causal masking in RetNet to exploit all the information within the current chunk and propose Non-Causal RetNet (NC-RetNet). Formally, the new D in the chunkwise recurrent representation is given by Eq. (4). The new masking is a full matrix instead of a lower triangular matrix. When predicting the 3D pose of the n^{th} frame, we can calculate the exponential decay of both past frames ($m < n$) and future frames ($m > n$)

within the chunk by using the absolute distance $jn - mj$ between the two frames. The mathematical expression of the new D matrix in the parallel representation can be found in the Supp. Mat. The chunk size, denoted by T, can be adjusted to balance the accuracy and inference latency. The larger T is, the more future information can be perceived by the model, but the longer the inference latency will be.

$$D = fD_{nm}g = f^{\,jn - mj}g; n; m \; 2 \; f1; :::; Tg \tag{4}$$

Fig. 2(right) illustrates the masks in the parallel and chunkwise recurrent representations of this non-causal retention. Note that the model can also be trained in parallel by using the staircase-shaped mask, but it does not have the recurrent representation unless $T = 1$.

By using the non-causal masking, NC-RetNet exploits temporal dependencies from the cross-chunk state which provides long-term historical information, and only a few future frames which provides some future information. Therefore, the temporal receptive field of our method is not limited by the chunk size. In fact, the chunk size in our method only affects the amount of future information while historical information is always adequate due to the cross-chunk state.

3.3 Transfer Knowledge from Large Chunks

We further develop a strategy for our NC-RetNet to improve its performance under small test chunk sizes, which is to train the model with a large chunk size and infer with a small chunk size. Since the model uses xPos, a relative position embedding, it is able to handle 2D sequences of different lengths from the form. In addition, the cross-chunk state S_i in the chunkwise recurrent representation is updated every chunk, containing a lot of information from previous chunks. With this long-term historical information, the model becomes insensitive to the length of future frames. Therefore, using a smaller chunk size during inference does not significantly decrease accuracy, but greatly reduces inference latency. This indicates that some knowledge is transferred from large chunks when training to the small chunks during inference.

Algorithm 1 Pseudo-code for training

Input: Training dataloader, initialized model, training chunk size T_l
Output: model after training

```
for input_2d, target in dataloader do
    L = input_2d.size(1) # Total length of the input sequence
    D_parallel = get_D_parallel(L, T_l) # Get the staircase-shaped mask for parallel training given the total length and chunk size
    pred = model.forward_parallel(input_2d, D_parallel)
    loss = loss_func(pred, target)
    loss.backward()
    optimizer.step()
end for
```

Algorithm 2 Pseudo-code for inference

Input: 2D stream, trained model, test chunk size T_s
Output: 3D stream {y_n}

```
D_chunkwise = get_D_chunkwise(T_s) # Get the full mask for chunkwise inference given the test chunk size
s_n, x_n = None, []
for x_i in stream do
  x_n.append(x_i)
  if len(x_n) == T_s then
    y_n, s_n = model.forward_chunkwise(x_n, D_chunkwise, s_n, n)
    x_n = []
    output(y_n) # Output y_n every chunk for downstream task
  end if
end for
```

The details of the training and testing are elaborated below. During training, we utilize RetNet's parallel representation to achieve training parallelism. We set the training chunk size to a large number T_l, to capture long-term patterns of human motion. The training pseudo-code is as shown in Algorithm 1. The parallel representation used during training implicitly incorporates the cross-chunk state. This means that the model can theoretically observe historical information over a long period of time as well as many future frames. During inference, the test chunk size T_s is set smaller than the training chunk size T_l and the chunkwise-recurrent representation is used. The pseudo-code for inference is as in Algorithm 2. Given a stream of 2D keypoints, the model processes the stream in the chunkwise-recurrent representation every T_s frames based on the current chunk x_n as well as an explicit cross-chunk state s_n. This cross-chunk state contains information about previous chunks and makes the model insensitive to the number of future frames. Therefore, although the chunk is smaller than the training chunks, the model can still extract stable temporal features.

3.4 Implementation Details

We implement our idea based on the state-of-the-art seq2seq method, MixSTE [46], by replacing its temporal encoder with RetNet. Since the movement of distal joints is more erratic than that of torso joints, the estimation of these distal joints should rely on more local temporal information. Therefore, we assign different decay coefficients to different human joints, which is referred to as joint-related decay coefficients. The chunkwise recurrent representation of it can be formulated in Eq. (5), where p is the index of human joints.

$$\begin{aligned} S_{i,p} &= K^T_{[i],p}(V_{[i],p} \quad {}_p) + {}^T_p S_{i\ \ 1,p} \\ Inner_{i,p} &= (Q_{[i],p} K^T_{[i],p} \quad D_p) V_{[i],p} \\ Cross_{i,p} &= (Q_{[i],p} S_{i\ \ 1,p}) \quad {}_p \\ \text{Retention}(X_{[i]}) &= Concat(Inner_{i,p} + Cross_{i,p}) \end{aligned} \tag{5}$$

The loss function and training strategies are the same as in MixSTE. The training chunk size is 243 on the Human3.6M dataset and 81 on the MPI-INF-3DHP dataset, and then we test the model with different chunk sizes to get the results. The overall architecture of our model is given in the Supp. Mat. We also implement our idea on MotionBERT [51] and the results can be found in the Supp. Mat.

4 Experiments

4.1 Datasets and Evaluation Protocols

Experiments are conducted on two human pose estimation datasets: Human3.6M [14] and MPI-INF-3DHP [23]. **Human3.6M** is the most widely used indoor dataset for single-person 3D human pose estimation, containing about 3.6 million images collected from 11 professional actors. Following the common practice [2, 27, 46, 50], we use the samples of S1, S5, S6, S7, S8 for training and evaluate on S9 and S11 subjects. Mean Per-Joint Position Error (MPJPE) and Procrustes-Aligned MPJPE (PA-MPJPE) are evaluated on this dataset. We also report the Mean Per-Joint Velocity Error (MPJVE) results, which reflect the continuity of the predicted results. **MPI-INF-3DHP** is a more challenging 3D human pose estimation dataset because it includes both indoor and outdoor scenes. The samples are collected from 8 subjects, each performing 8 actions. The test set consists of 6 subjects in different scenes. We follow the setup in [2, 39, 46, 50]. For the MPI-INF-3DHP dataset, we report the results of MPJPE, Percentage of Correct Keypoints (PCK) within the 150mm range, and Area Under Curve (AUC), following [18, 38, 50].

4.2 Quantitative Comparison

Results on Human3.6M We first use the 2D keypoints detected by CPN [3] as the input, and the results are shown in Tab. 1. It can be seen that our method achieves comparable performance with the state-of-the-art when the chunk size for inference is large ($T = 243$). Moreover, our method outperforms previous methods by a clear margin when the chunk size is small ($T = 27$, 81), as the accuracy of our method only decreases slightly when the chunk size is reduced. Furthermore, the MPJPE at $T = 27$ is comparable to the previous state-of-the-art method at $T = 81$ (42.1 mm vs. 42.0 mm). This indicates that our method can provide similarly accurate predictions with much lower inference latency. Additionally, our method exhibits significant improvement in continuity compared with previous methods, with a 0.2 mm per frame improvement on the MPJVE metric.

We further use the ground truth 2D keypoints of the Human3.6M dataset as input to test the upper bound of our method, as shown in Tab. 2. The results indicate that using the model trained with 243 frames to infer with a chunk size of 81 improves the MPJPE metric by 2.3 mm compared to the previous state-of-the-art (22.4 mm vs. 25.7 mm). Moreover, the MPJPE of our method at $T = 27$

is remarkably lower than that of previous methods at $T = 81$, demonstrating our method's efficient utilization of transferred knowledge to achieve higher accuracy with lower inference latency.

Table 1: Comparison of MPJPE, PA-MPJPE and MPJVE on the Human3.6M dataset using 2D keypoints detected by CPN [3] as input. T is the chunk size when testing.

MPJPE	T	Dir.	Disc.	Eat.	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Average
MixSTE [46]	27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	45.1
STCFormer [38]	27	40.7	44.6	41.2	41.9	45.8	53.7	41.5	40.9	55.9	63.8	44.6	41.5	44.7	29.5	30.8	44.1
Ours	27	38.0	41.5	40.0	40.0	44.1	51.3	39.8	41.7	53.1	58.3	43.5	39.8	42.0	28.4	29.6	**42.1**
Anatomy [2]	81	42.1	43.8	41.0	43.8	46.1	53.5	42.4	43.1	53.9	60.5	45.7	42.1	46.2	32.2	33.8	44.6
PoseFormer [50]	81	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
Xue *et al.* [43]	81	42.1	45.3	40.9	42.9	45.4	52.7	42.6	42.5	55.3	61.8	44.9	41.7	44.9	29.9	30.8	44.2
P-STMO [31]	81	41.7	44.5	41.0	42.9	46.0	51.3	42.8	41.3	54.9	61.8	45.1	42.8	43.8	30.8	30.7	44.1
MixSTE [46]	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
STCFormer [38]	81	40.6	43.0	38.3	40.2	43.5	52.6	40.3	40.1	51.8	57.7	42.8	39.8	42.3	28.0	29.5	42.0
Ours	81	36.9	40.5	39.0	38.6	43.3	49.6	38.8	40.2	52.6	56.5	42.6	38.8	40.5	26.8	28.4	**40.9**
VideoPose3D [27]	243	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
Anatomy [2]	243	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
Xue *et al.* [43]	243	39.9	42.7	40.3	42.3	45.0	52.8	40.4	39.3	56.9	61.2	44.1	41.3	42.8	28.4	29.3	43.1
MHFormer [18]	351	39.2	43.1	40.1	40.9	44.9	51.2	40.6	41.3	53.5	60.3	43.7	41.1	43.8	29.8	30.6	43.0
P-STMO [31]	243	38.9	42.7	40.4	41.1	45.6	49.7	40.9	39.9	55.5	59.4	44.9	42.2	42.7	29.4	29.4	42.8
MixSTE [46]	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
STCFormer [38]	243	38.4	41.2	36.8	38.0	42.7	50.5	38.7	38.2	52.5	56.8	41.8	38.4	40.2	26.2	27.7	40.5
Ours	243	36.9	40.1	38.7	38.3	42.9	48.6	38.2	40.0	52.5	55.4	42.3	38.7	39.7	26.2	27.8	**40.4**

PA-MPJPE	T	Dir.	Disc.	Eat.	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Average
STCFormer [38]	27	31.9	35.1	32.7	34.1	34.9	41.3	32.1	31.6	45.0	50.6	36.0	31.7	35.5	23.6	25.1	34.8
Ours	27	31.7	33.9	32.3	33.3	35.2	39.1	31.0	31.9	44.0	48.7	36.0	31.0	34.6	23.0	24.8	**34.0**
Anatomy [2]	81	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
PoseFormer [50]	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
Xue *et al.* [43]	81	31.6	35.5	32.3	34.2	35.1	40.3	32.3	32.3	44.5	49.6	35.8	31.6	35.0	23.7	24.7	34.6
MixSTE [46]	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
STCFormer [38]	81	30.4	33.8	31.1	31.7	33.5	39.5	30.8	30.0	41.8	45.8	34.3	30.1	32.8	21.9	23.4	32.7
Ours	81	30.5	33.1	31.4	31.6	33.0	38.4	29.8	30.6	43.6	45.4	34.4	30.3	32.4	21.5	22.2	**32.6**
VideoPose3D [27]	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
Anatomy [2]	243	32.6	35.1	32.8	35.4	36.3	40.4	32.4	32.3	42.7	49.0	36.8	32.4	36.0	24.9	26.5	35.0
Xue *et al.* [43]	243	31.2	34.1	31.9	33.8	33.9	39.5	31.6	30.0	45.4	48.1	35.0	31.1	33.5	22.4	23.6	33.7
P-STMO [31]	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
MixSTE [46]	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
STCFormer [38]	243	29.3	33.0	30.7	30.6	32.7	38.2	29.7	28.8	42.2	45.0	33.3	29.4	31.5	20.9	22.3	**31.8**
Ours	243	30.8	33.1	31.3	31.8	33.4	37.7	30.1	30.5	43.4	45.5	34.3	30.3	31.5	21.4	22.7	32.5

MPJVE	T	Dir.	Disc.	Eat.	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Average
VideoPose3D [27]	243	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
Anatomy [2]	243	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
PoseFormer [50]	81	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
StridedFormer [17]	351	2.4	2.5	1.8	2.8	1.8	2.2	2.2	2.5	1.5	2.0	1.8	1.9	3.2	2.5	2.1	2.2
MixSTE [46]	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
Ours	81	2.3	2.4	1.8	2.6	1.7	2.1	2.1	2.5	1.5	2.1	1.8	1.9	3.0	2.4	2.0	2.2
Ours	243	2.3	2.4	1.8	2.6	1.7	2.1	2.1	2.5	1.5	2.1	1.8	1.9	3.0	2.4	2.0	**2.0**

Results on MPI-INF-3DHP The results on the MPI-INF-3DHP datasets are shown in Tab. 3. An improvement of 0.9 mm on the MPJPE metric is achieved at $T = 81$, and the improvement becomes more remarkable as the chunk size decreases. In particular, our method outperforms previous methods very significantly with an improvement of 4.1 mm on the MPJPE metric when T is 9. Similar to the phenomenon on the Human3.6M dataset, our method is

Table 2: Comparison of MPJPE on the Human3.6M dataset using 2D ground truth keypoints as input.

MPJPE	T	Dir.	Disc.	Eat.	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Average
Ours	27	23.7	24.8	23.5	24.4	23.6	28.1	27.2	25.3	26.7	27.9	25.0	23.5	23.6	17.2	18.7	**24.2**
PoseFormer [50]	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
Xue *et al.* [43]	81	27.6	28.8	24.9	25.7	26.7	30.6	30.8	26.4	35.8	32.7	27.1	26.2	25.6	19.2	20.6	27.2
MixSTE [46]	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
STCFormer [38]	81	26.2	26.5	23.4	24.6	25.0	28.6	28.3	24.6	30.9	33.7	25.7	25.3	24.6	18.6	19.7	25.7
Ours	81	20.9	22.5	21.8	21.5	22.0	25.6	23.4	23.7	28.1	28.8	23.9	20.9	21.1	14.9	16.3	**22.4**
Xue *et al.* [43]	243	25.8	25.2	23.3	23.5	24.0	27.4	27.9	24.4	29.3	30.1	24.9	24.1	23.3	18.6	19.7	24.7
MHFormer [18]	351	27.7	32.1	29.1	28.9	30.0	33.9	33.0	31.2	37.0	39.3	30.0	31.0	29.4	22.2	23.0	30.5
P-STMO [31]	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
MixSTE [46]	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
STCFormer [38]	243	21.4	22.6	21.0	21.3	23.8	26.0	24.2	20.0	28.9	28.0	22.3	21.4	20.1	14.2	15.0	22.0
Ours	243	20.0	21.1	20.9	20.8	20.1	24.9	23.5	22.5	26.5	39.6	21.7	20.9	20.4	14.5	15.7	**21.5**

comparable to or better than previous methods with an even smaller chunk size. For example, the MPJPE of our method at $T = 27$ is better than STCFormer at $T = 81$ (22.7 mm vs. 23.1 mm). And the MPJPE of our method at $T = 9$ is similar to STCFormer at $T = 27$ (24.1 mm vs. 24.2 mm). These results show that our method generalizes well on different datasets.

Table 3: Comparison of quantitative results on the MPI-INF-3DHP dataset. ↑: higher is better. ↓: lower is better.

Method	T	PCK↑	AUC↑	MPJPE↓
PoseFormer [50]	9	88.6	56.4	77.1
CrossFormer [10]	9	89.1	57.5	76.3
MHFormer [18]	9	93.8	63.3	58.0
STCFormer [38]	9	98.2	81.5	28.2
Ours	9	**98.9**	**83.3**	**24.1**
Lin *et al.* [20]	25	83.6	51.4	79.8
MixSTE [46]	27	94.4	66.5	54.9
STCFormer [38]	27	98.4	83.4	24.2
Ours	27	**99.1**	**84.1**	**22.7**
UGCN [39]	96	86.9	62.1	68.1
Anatomy [2]	81	87.8	53.8	79.1
Hu *et al.* [12]	96	97.9	69.5	42.5
Einfalt *et al.* [8]	81	95.4	67.6	46.9
P-STMO [31]	81	97.9	75.8	32.2
STCFormer [38]	81	98.7	83.9	23.1
Ours	81	**99.1**	**84.4**	**22.2**

4.3 Qualitative Results

Visualization on Continuity We compute the MPJVE of the results predicted by MixSTE, STCFormer and our method at different timesteps, and visualize the curves in Fig. 3. It can be seen that the MPJVE of our method is lower than

that of previous methods. Our method captures temporal information using non-overlapping shift windows, similar to MixSTE. However, our method produces more continuous results at the edge between two chunks compared to MixSTE. MixSTE independently estimates two consecutive chunks, which results in a lack of continuity at the edge. n contrast, our method incorporates temporal information from previous chunks through the cross-chunk state, which improves the continuity. Compared to STCFormer, our method generally produces more continuous results. This is because our method generates multiple frames each time, allowing for the continuity constraints to the output.

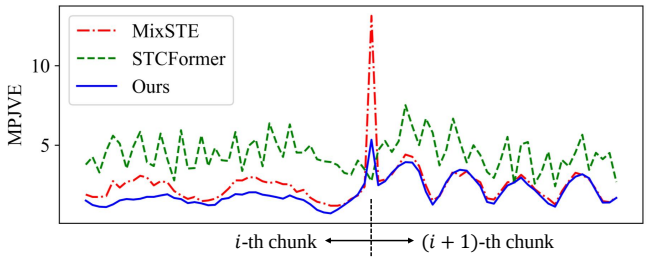

Fig. 3: Comparison of the MPJVE curves over time between MixSTE, STCFormer and our method.

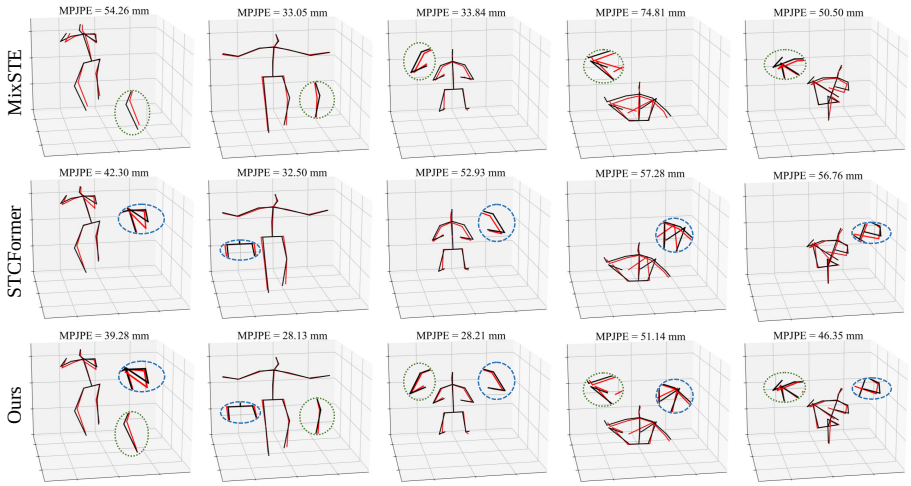

Fig. 4: Comparison of some visualization results predicted by MixSTE [46], STCFormer [38] and our method. The black skeletons are the ground truth, and the red skeletons are the predicted results. The comparison with MixSTE is shown in green circles, while the comparison with STCFormer is shown in blue circles.

Visualization of Results We present some visualization examples in Fig. 4, where the results are predicted by MixSTE [46], STCFormer [38] and our method, respectively. It can be seen that our method predicts more accurate results, and the improvement is visually obvious. More visualization results can be found in the Supp. Mat.

4.4 Ablation Study

Ablations on Knowledge Transfer The knowledge transferred from large training chunks to smaller test chunks plays an important role in our method. To demonstrate this, we train the model with chunk sizes of 27 and 81, respectively, and compare the performance of these models with that of the model trained with $T = 243$. The results are shown in Tab. 4 (2^{nd} to 4^{th} rows). It can be seen that compared with the models trained with $T = 27$ and 81, using the models trained with a larger chunk size ($T = 243$) for inference is significantly better. This indicates that the knowledge learned with large chunks is useful for reasoning about small chunks.

Table 4: Comparison of different methods in terms of knowledge transfer.

Method	Train T	Test $T = 27$	Test $T = 81$	Test $T = 243$
Previous SOTA	Same as test T	44.1	42.0	40.5
Ours	27	43.7	-	-
Ours	81	43.0	41.9	-
Ours	243	**42.1**	**40.9**	**40.4**
MixSTE w.t. xPos	27	45.3	-	-
MixSTE w.t. xPos	81	47.0	42.6	-
MixSTE w.t. xPos	243	48.8	44.1	41.1
Ours w.o. state	27	46.3	-	-
Ours w.o. state	81	49.1	44.0	-
Ours w.o. state	243	54.2	49.8	42.5

Effect of Cross-Chunk State We compare our methods with two baselines that do not use the cross-chunk state: the MixSTE model with xPos as the position embedding, and the model based on RetNet but without the cross-chunk state. These two baselines are able to handle sequences of different lengths, but can only use within-chunk information. The results are shown in Tab. 4 (bottom six rows). It can be seen that the two baselines without long-term historical information deteriorate rapidly as the gap between the training and test chunk sizes increases. This means that they cannot efficiently transfer knowledge from large chunks to small chunks. Therefore, the cross-chunk state is essential for knowledge transfer in our method, and our NC-RetNet is the first method to have this knowledge transfer property.

Comparison of Computational Cost The comparison of the model parameters and computational cost of our method and previous methods is as shown in Tab. 5. For seq2seq methods, the FLOPs are averaged over the number of frames, since the prediction of a single inference yields results over multiple frames. It can be seen that our modification of MixSTE does not bring any increase in the model parameters or FLOPs. And compared to STCFormer [38], which has comparable performance to our method at $T = 243$, the computational cost of our method is much lower (430 M vs. 78107 M).

Table 5: Comparison of model parameters, computational cost. FLOPs for seq2seq and our methods is averaged over the number of output frames, as is done in [46].

Method	Params (M)	FLOPs (M)	MPJPE (T=243)
StridedFormer [17]	4.2	1372	44.0
P-STMO [31]	6.7	1737	42.8
MHFormer [18]	24.7	4812	43.2
MixSTE [46]	33.6	572	40.9
STCFormer [38]	18.9	78107	40.5
Ours	25.2	430	40.4

5 Conclusion

In this paper, we propose the first 3D human pose estimation model based on Retentive Networks, NC-RetNet. By using the non-causal masking, it effectively leverages a large number of past frames and a limited number of future frames to incorporate temporal information. Furthermore, we introduce a knowledge transfer strategy that involves training the model with a larger chunk size and using a smaller chunk size during inference, resulting in reduced inference latency without too much loss in accuracy. Through extensive experiments on the Human3.6M and MPI-INF-3DHP datasets, our approach has demonstrated state-of-the-art performance even with a smaller test chunk size. In conclusion, our method achieves a good balance between high accuracy and low inference latency, making it suitable for real-time scenarios.

Limitations Admittedly, there are two limitations in our work. Firstly, the fundamental theory behind our method's ability to transfer knowledge is unclear, despite our study of the effect of the cross-chunk state. Secondly, we have only tested our method in the 2D-to-3D lifting task. However, the idea of transferring knowledge from large chunks to smaller chunks is universal to many sequential data in computer vision. Further work is required to explain the theory and explore more applications.

References

1. Cai, Y., Ge, L., Liu, J., Cai, J., Cham, T.J., Yuan, J., Thalmann, N.M.: Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2272–2281 (2019) 2, 4
2. 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 **32**(1), 198–209 (2021) 1, 9, 10, 11
3. 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, 9, 10
4. Chi, T.C., Fan, T.H., Ramadge, P.J., Rudnicky, A.: Kerple: Kernelized relative positional embedding for length extrapolation. Advances in Neural Information Processing Systems **35**, 8386–8399 (2022) 5
5. Choi, H., Moon, G., Chang, J.Y., Lee, K.M.: Beyond static features for temporally consistent 3d human pose and shape from a video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1964–1973 (2021) 4
6. Choi, S., Choi, S., Kim, C.: Mobilehumanpose: Toward real-time 3d human pose estimation in mobile devices. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2328–2338 (2021) 4
7. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014) 4
8. 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
9. Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C.: Ghostnet: More features from cheap operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1580–1589 (2020) 4
10. Hassanin, M., Khamiss, A., Bennamoun, M., Boussaid, F., Radwan, I.: Crossformer: Cross spatio-temporal transformer for 3d human pose estimation. arXiv preprint arXiv:2203.13387 (2022) 4, 11
11. Hesse, N., Schröder, A.S., Müller-Felber, W., Bodensteiner, C., Arens, M., Hofmann, U.G.: Body pose estimation in depth images for infant motion analysis. In: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 1909–1912. IEEE (2017) 1
12. Hu, W., Zhang, C., Zhan, F., Zhang, L., Wong, T.T.: Conditional directed graph convolution for 3d human pose estimation. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 602–611 (2021) 11
13. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. arXiv preprint arXiv:1602.07360 (2016) 4
14. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: 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 (2013) 3, 9

15. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016) 3, 4
16. Li, H., Shi, B., Dai, W., Zheng, H., Wang, B., Sun, Y., Guo, M., Li, C., Zou, J., Xiong, H.: Pose-oriented transformer with uncertainty-guided refinement for 2d-to-3d human pose estimation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 1296–1304 (2023) 4
17. 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 **25**, 1282–1293 (2022) 4, 10, 14
18. Li, W., Liu, H., Tang, H., Wang, P., Van Gool, L.: Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13147–13156 (2022) 4, 9, 10, 11, 14
19. Li, Y., Zhang, S., Wang, Z., Yang, S., Yang, W., Xia, S.T., Zhou, E.: Tokenpose: Learning keypoint tokens for human pose estimation. In: Proceedings of the IEEE/CVF International conference on computer vision. pp. 11313–11322 (2021) 3
20. Lin, J., Lee, G.H.: Trajectory space factorization for deep video-based 3d human pose estimation. arXiv preprint arXiv:1908.08289 (2019) 11
21. 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) 3
22. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3d human pose estimation. In: Proceedings of the IEEE international conference on computer vision. pp. 2640–2649 (2017) 1, 3
23. Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., Theobalt, C.: Monocular 3d human pose estimation in the wild using improved cnn supervision. In: 2017 international conference on 3D vision (3DV). pp. 506–516. IEEE (2017) 3, 9
24. Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H.P., Xu, W., Casas, D., Theobalt, C.: Vnect: Real-time 3d human pose estimation with a single rgb camera. Acm transactions on graphics (tog) **36**(4), 1–14 (2017) 4
25. 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
26. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Coarse-to-fine volumetric prediction for single-image 3d human pose. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7025–7034 (2017) 3
27. 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) 2, 4, 9, 10
28. Press, O., Smith, N.A., Lewis, M.: Train short, test long: Attention with linear biases enables input length extrapolation. arXiv preprint arXiv:2108.12409 (2021) 5
29. Rayat Imtiaz Hossain, M., Little, J.J.: Exploiting temporal information for 3d pose estimation. arXiv e-prints pp. arXiv–1711 (2017) 2

30. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018) 4
31. 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
32. Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. arXiv preprint arXiv:2104.09864 (2021) 4
33. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: CVPR (2019) 3
34. Sun, X., Shang, J., Liang, S., Wei, Y.: Compositional human pose regression. In: Proceedings of the IEEE international conference on computer vision. pp. 2602–2611 (2017) 3
35. Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., Wei, F.: Retentive network: A successor to transformer for large language models. arXiv preprint arXiv:2307.08621 (2023) 2, 5
36. Sun, Y., Dong, L., Patra, B., Ma, S., Huang, S., Benhaim, A., Chaudhary, V., Song, X., Wei, F.: A length-extrapolatable transformer. arXiv preprint arXiv:2212.10554 (2022) 4
37. Svenstrup, M., Tranberg, S., Andersen, H.J., Bak, T.: Pose estimation and adaptive robot behaviour for human-robot interaction. In: 2009 IEEE International Conference on Robotics and Automation. pp. 3571–3576. IEEE (2009) 1
38. 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
39. 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) 1, 9, 11
40. Wehrbein, T., Rudolph, M., Rosenhahn, B., Wandt, B.: Probabilistic monocular 3d human pose estimation with normalizing flows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11199–11208 (2021) 1
41. Xu, T., Takano, W.: Graph stacked hourglass networks for 3d human pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16105–16114 (2021) 1
42. Xu, Y., Zhang, J., Zhang, Q., Tao, D.: Vitpose: Simple vision transformer baselines for human pose estimation. Advances in Neural Information Processing Systems **35**, 38571–38584 (2022) 3
43. Xue, Y., Chen, J., Gu, X., Ma, H., Ma, H.: Boosting monocular 3d human pose estimation with part aware attention. IEEE Transactions on Image Processing **31**, 4278–4291 (2022) 2, 10, 11
44. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018) 1
45. Zeng, A., Sun, X., Yang, L., Zhao, N., Liu, M., Xu, Q.: Learning skeletal graph neural networks for hard 3d pose estimation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11436–11445 (2021) 1
46. 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

47. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6848–6856 (2018) 4
48. Zhao, Q., Zheng, C., Liu, M., Chen, C.: A single 2d pose with context is worth hundreds for 3d human pose estimation. Advances in Neural Information Processing Systems **36** (2024) 1, 3
49. 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
50. 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) 2, 4, 9, 10, 11
51. 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