KTPFormer: Kinematics and Trajectory Prior Knowledge-Enhanced Transformer for 3D Human Pose Estimation

Jihua Peng[1,2] Yanghong Zhou[1] P. Y. Mok[1,2,*]
[1]The Hong Kong Polytechnic University [2]Laboratory for Artificial Intelligence in Design
{ji-hua.peng,yanghong.zhou}@connect.polyu.hk, tracy.mok@polyu.edu.hk

Abstract

This paper presents a novel Kinematics and Trajectory Prior Knowledge-Enhanced Transformer (KTPFormer), which overcomes the weakness in existing transformer-based methods for 3D human pose estimation that the derivation of Q, K, V vectors in their self-attention mechanisms are all based on simple linear mapping. We propose two prior attention modules, namely Kinematics Prior Attention (KPA) and Trajectory Prior Attention (TPA) to take advantage of the known anatomical structure of the human body and motion trajectory information, to facilitate effective learning of global dependencies and features in the multi-head self-attention. KPA models kinematic relationships in the human body by constructing a topology of kinematics, while TPA builds a trajectory topology to learn the information of joint motion trajectory across frames. Yielding Q, K, V vectors with prior knowledge, the two modules enable KTPFormer to model both spatial and temporal correlations simultaneously. Extensive experiments on three benchmarks (Human3.6M, MPI-INF-3DHP and HumanEva) show that KTPFormer achieves superior performance in comparison to state-of-the-art methods. More importantly, our KPA and TPA modules have lightweight plug-and-play designs and can be integrated into various transformer-based networks (i.e., diffusion-based) to improve the performance with only a very small increase in the computational overhead. The code is available at: https://github.com/JihuaPeng/KTPFormer.

1. Introduction

3D human pose estimation aims to predict the 3D coordinates of human body joints from the input of either monocular 2D images or video sequences. Because of the strong ties to action recognition [20, 21], human-robot interaction [13, 33] and virtual reality [9], the research of 3D human pose estimation has received considerable attention in recent years. Transformer, a deep learning architecture, has revolutionized first in natural language processing (NLP) and later in other areas such as computer vision since its introduction in 2017 [36]. The name ’transformer’ comes from the fact that these architectures use a self-attention mechanism to transform layers of inputs into layers of outputs in a way that allows the model to focus on (attend to) certain inputs. In terms of 3D pose estimation, the transformer first processes an input video into a sequence of tokens, the basic units of processing namely 2D poses, and then models the spatial-temporal relationship between tokens using *multi-head self-attention* (MHSA) mechanism.

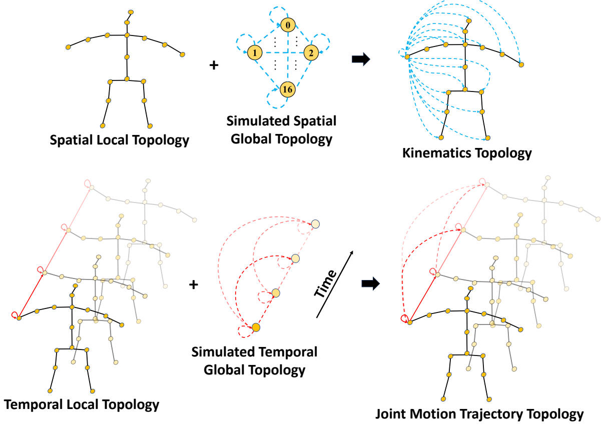

0
1
2
16
+
Spatial Local Topology
Simulated Spatial Global Topology
Kinematics Topology
Time
Temporal Local Topology
Simulated Temporal Global Topology
Joint Motion Trajectory Topology

Figure 1. **Top**: the spatial local topology (fixed) plus the simulated spatial global topology (learnable) to form the kinematics topology (learnable). **Bottom**: the temporal local topology (fixed) plus the simulated temporal global topology (learnable) to form the joint motion trajectory topology (learnable).

The existing works of transformer-based methods for 3D human pose estimation [17, 18, 30, 34, 42, 44, 46] mainly focus on developing novel *transformer encoders*. They model either the spatial correlation between joints within each frame and the pose-to-pose or joint-to-joint temporal correlation across frames. Regardless of spatial or temporal MHSA calculation, the present transformer-based methods all use linear embedding where 2D pose sequence are tokenized into high dimensional features and treated uniformly

[*]Corresponding author

to compute the spatial correlation between joints and the temporal correlation across frames in the spatial and temporal MHSA, respectively. This may lead to the problem of ’attention collapse’, a phenomenon denoting a circumstance wherein the self-attention becomes too focused on a limited subset of input tokens while disregarding other segments of the sequence. In contrast to previous works, with the known anatomical structure of the human body as well as joint motion trajectory across frames as a priori knowledge, we propose a graph-based method to formulate such prior knowledge-attention for better learning the spatial and temporal correlations. Our graph-based prior attention mechanism is different from other existing graph-transformer methods [7, 14, 45, 47]; without modifying the transformer structure or introducing complex network, instead, we design plug-and-play modules to be placed in front of MHSA modules of a vanilla transformer. Our method is simple yet effective, highly flexible and adaptable, allowing it to be integrated into different transformer-based methods.

To be specific, we introduce two novel prior attention modules, namely Kinematics Prior Attention (KPA) and Trajectory Prior Attention (TPA), and the key concepts are illustrated in Figure 1. KPA first constructs a spatial *local* topology based on the anatomy of the human body, as shown at the top of Figure 1. The way these joints are physically connected to each other is fixed and is represented by solid lines. To introduce the kinematic relations among non-connected joints, we use a fully connected spatial topology to calculate the joint-to-joint attention weights, called simulated spatial global topology. In this topology, the strength of the connectivity relationship between each joint (including itself) is learnable, and thus we denote it with a dotted line in Figure 1. We combine the spatial local topology and the simulated spatial global topology to obtain a *kinematics topology*, where each joint has a learnable kinematic relationship with each other. This kinematic topological information aims to provide a priori knowledge to the spatial MHSA, enabling it to assign weights to joints based on the kinematic relationships in different actions. Similarly, as shown in the bottom of Figure 1, TPA connects the same joint across consecutive frames to build the temporal *local* topology. Next, we construct a temporal global topology by exploiting learnable vectors (dotted line) to connect the joints among all neighboring and non-neighboring frames, which is equivalent to the computation of attention weights among all frames by self-attention, called simulated temporal global topology. Then, we combine the two topologies to obtain a new topology called *joint motion trajectory topology*, which allows the network to learn both the temporal sequentiality and periodicity (joints in non-neighboring frames have similar motions to each other) for the joint motion. The temporal tokens embedded with the trajectory information will be more effectively activated in the temporal MHSA, which enhances the temporal modeling ability for MHSA. The KPA and TPA modules are combined with vanilla MHSA and MLP to form the Kinematics and Trajectory Prior Knowledge-Enhanced Transformer (KTPFormer) for 3D pose estimation, as shown in Figure 2. In summary, the contributions of this paper are as follows:

- We propose two novel prior attention modules, KPA and TPA, which can be combined with MHSA and MLP in a simple yet effective way, forming the KTPFormer for 3D pose estimation.
- Our KTPFormer outperforms the state-of-the-art methods on Human3.6M, MPI-INF-3DHP and HumanEva benchmarks, respectively.
- KPA and TPA are designed as lightweight plug-and-play modules, which can be integrated into various transformer-based methods (including diffusion-based) for 3D pose estimation. Extensive experiments show that our method can effectively improve the performance without largely increasing computational resources.

2. Related Work

Monocular 3D human pose estimation methods can be broadly categorized into two approaches: (1) direct 3D methods [16, 25, 27, 28, 35, 38] and (2) 2D-to-3D lifting methods [2, 10, 22, 23, 29]. The second approach first estimates 2D joint coordinates from input image using 2D human pose detectors [3, 26], next they are lifted into 3D space. These methods can be further categorized as transformer-based or graph-based.

Transformer-based networks. Transformer was first proposed by Vaswani et al. [36] and showed remarkable performance in natural language processing (NLP), as the self-attention can model long-range dependencies and also capture global features. Recently, several studies on transformer-based methods for 3D human pose estimation have been reported, with Poseformer [46] being the first that predicts the 3D pose of the central frame by modeling spatial and temporal information. However, the computational burden is huge when the frame number increases. PoseformerV2 [44] introduces a time-frequency feature to the transformer structure, efficiently extends the input sequence length, and achieves a good trade-off between speed and accuracy. MHFormer [18], a transformer-based network, generates multiple hypotheses at the pose level and calculates the target 3D pose by averaging. MixSTE [42] stacks spatial and temporal transformer blocks to capture spatial-temporal features alternatively and models the trajectory of joints over frame sequence. STCFormer [34] slices the input joint features into two partitions and uses MHSA to encapsulate the spatial and temporal context in parallel. D3DP [31], a diffusion-based method, recovers the noisy 3D poses by assembling joint-by-joint multiple hypotheses. By introducing new encoders for better modeling the spatial and temporal relations, these methods all have unavoidably changed the internal structure or altered the MHSA of the transformer, resulting in largely increased network complexity.

Graph networks and graph-transformer methods. With a strong capability of capturing dynamic relations, graph neural networks are widely used in 3D human pose estimation [1, 4, 5, 11, 19, 37, 39, 40, 43, 48]. Zhao et al. [43] introduced a semantic graph convolutional network (GCN) that involved a learnable mask to scale up the receptive field of convolution filters, capturing semantic information among local and global nodes. Ci et al. [5] proposed a locally connected network (LCN) to enhance the representation capability over GCN. Xu and Takano [39] developed a graph stacked hourglass network to learn different scales of human skeletal representations. Yu et al. [40] utilized the adaptive GCN [15] to model global correlations and learn local joint representation by individually connected layers.

Recently, some studies combined graph and transformer, introducing graph-transformer methods [7, 14, 45, 47]. PoseGTAC [47] uses graph atrous convolution to learn the multi-scale information among 1-to-3 top neighbors and utilizes the graph transformer layer to capture long-range features. GraFormer [45] replaces the MLP in the transformer with learnable GCN layers to form the GraAttention block, which also contains MHSA. ChebGConv [6] models the implicit connection relations among non-neighboring joints. Li et al. [14] introduces a graph POT, where each element is the relative distance between a pair of joints, which are being encoded as the attention bias in the MHSA module. DiffPose [7] interlaces GCN layers [43] with self-attention layers as a diffusion model, which can capture spatial features between joints based on the human skeleton. Nevertheless, these graph-transformer methods [7, 14, 45, 47] learn merely the spatial information of individual pose, without considering temporal correlation across frames. Moreover, they [7, 14, 45, 47] modify the structure of the transformer by introducing the graph convolution, resulting in much larger and more complex networks.

3. Method

In this paper, we propose a novel Kinematics and Trajectory Prior Knowledge-Enhanced Transformer (KTPFormer), which combines kinematics and trajectory prior attentions and MHSA in a direct but effective way. Our KTPFormer can model both spatial and temporal information simultaneously. Moreover, our method preserves the inherent structure of the transformer and is more flexible.

Our KTPFormer utilizes the seq2seq pipeline for 3D human pose estimation, which can simultaneously predict 3D pose sequence corresponding to the input 2D keypoint sequence. As shown in Figure 2, an input 2D pose sequence $P_{TN} \in \mathbb{R}^{T \times N \times 2}$ is first fed into the Kinematics-Enhanced Transformer, with T denotes the number of frames, N denotes the number of joints, and 2 is the channel size. KPA injects the kinematic topological information into the 2D pose $P_N \in \mathbb{R}^{N \times 2}$ in each frame, aiming to obtain high-dimensional spatial tokens $H_{TN} \in \mathbb{R}^{T \times N \times d_m}$. Next, the spatial MHSA transforms H_{TN} into matrics Q_S, K_S, V_S for learning the global correlation between joints. The Trajectory-Enhanced Transformer takes a sequence of reshaped tokens $P_{NT} \in \mathbb{R}^{N \times T \times d_m}$ as inputs. We stack two TPA blocks with the residual connection to generate the temporal tokens $H_{NT} \in \mathbb{R}^{N \times T \times d_m}$ with incorporated prior information on joint motion trajectories. Next, the temporal MHSA transforms H_{NT} into Q_T, K_T, V_T for modeling global coherence among frames. The output features from Temporal MHSA are reshaped and fed into stacked spatio-temporal transformers for encoding. Finally, the regression head predicts the coordinates of the 3D pose sequence based on the learned features.

3.1. Kinematics-Enhanced Transformer

Kinematics-Enhanced Transformer receives the input 2D keypoint sequence and transforms them into high-dimensional spatial tokens. The 2D keypoint sequence first goes through the KPA for embedding the prior knowledge of kinematics, which is then fed into the spatial MHSA for global correlation learning between joints.

To be specific, given a 2D pose sequence as $P_{TN} \in \mathbb{R}^{T \times N \times 2}$, we regard each joint $p_i \in \mathbb{R}^2$ of a 2D pose $P_N \in \mathbb{R}^{N \times 2}$ as a keypoint patch. Next, we define a learnable transformation matrix $W \in \mathbb{R}^{2 \times d_m}$ to map all keypoint patches P_{TN} into high-dimensional tokens $\bar{P}_{TN} \in \mathbb{R}^{T \times N \times d_m}$. In order to inject the prior information of kinematics into $\bar{P}_{TN}$, KPA first constructs a symmetric affinity matrix $A_N \in \mathbb{R}^{N \times N}$ based on the skeletal structure of the human body, namely spatial local topology, as shown in Figure 1. If two joints are physically connected in the human body structure, the corresponding element in the affinity matrix $A_N \in \mathbb{R}^{N \times N}$ is non-zero and 0 otherwise. The affinity matrix A_N can allow each 2D keypoint to learn anatomical structure information of the human body. Besides, KPA also considers the implicit kinematic relationships among adjacent and non-adjacent keypoints. Similar to the self-attention in MHSA, we establish a fully connected spatial topology, called simulated spatial global topology, as shown in Figure 1. In this topology, all the joints are interconnected by dotted lines, indicating that the connectivity relationship between each joint is learnable. The simulated spatial global topology is denoted as an affinity matrix $\hat{A}_N \in \mathbb{R}^{N \times N}$, where each element is learnable. Lastly, we combine the spatial local topology A_N with the simulated spatial global topology $\hat{A}_N$ to derive a kinematics

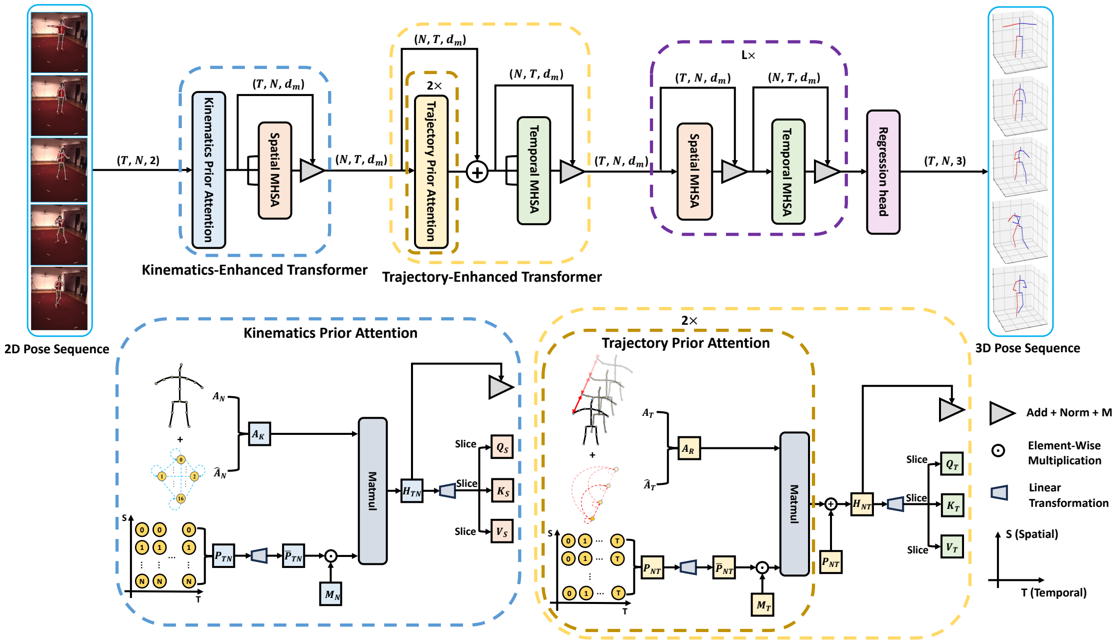


Figure 2. Overview of Kinematics and Trajectory Prior Knowledge-Enhanced Transformer (KTPFormer). The input 2D pose sequence $P_{TN} \in \mathbb{R}^{T \times N \times 2}$ with T frames and N joints is first fed into the Kinematics-Enhanced Transformer. KPA injects the kinematic information into the P_{TN}, aiming to obtain high-dimensional spatial tokens $H_{TN} \in \mathbb{R}^{T \times N \times d_m}$. Then, H_{TN} is split into Q_S, K_S, V_S, which are then fed into the Spatial MHSA. The Trajectory-Enhanced Transformer takes a sequence of reshaped tokens $P_{NT} \in \mathbb{R}^{N \times T \times d_m}$ as input. The stacked TPA blocks with the residual connection yield the temporal tokens $H_{NT} \in \mathbb{R}^{N \times T \times d_m}$, which are then sliced into Q_T, K_T, V_T for the Temporal MHSA.

topology A_K, which is shown as:

$$A_K = \frac{(A_N + \hat{A}_N) + (A_N + \hat{A}_N)'}{2} \tag{1}$$

where $'$ denotes the matrix transpose, $A_K \in \mathbb{R}^{N \times N}$ is a learnable affinity matrix. The reason why we construct the A_K with the above formula is that the original spatial local topology matrix A_N is also symmetric. In order to ensure that different keypoints can learn different kinematic knowledge, we introduce a learnable weight matrix $M_N \in \mathbb{R}^{N \times d_m}$ and multiply it with tokens $\bar{P}_{TN} \in \mathbb{R}^{T \times N \times d_m}$ by element-wise multiplication, which is an economic and effective way. Thus, we can obtain the tokens $H_{TN} \in \mathbb{R}^{T \times N \times d_m}$ including the prior knowledge of kinematics. The formula is represented as:

$$H_{TN} = (M_N \odot \bar{P}_{TN})A_K \tag{2}$$

where $\odot$ represents element-wise multiplication. Moreover, we add the learnable spatial positional embedding to H_{TN}. After that, $H_{TN} \in \mathbb{R}^{T \times N \times d_m}$ is transformed into queries $Q_S \in \mathbb{R}^{T \times N \times d_m}$, keys $K_S \in \mathbb{R}^{T \times N \times d_m}$ and values $V_S \in \mathbb{R}^{T \times N \times d_m}$ by a linear transformation matrix. Then, we design a spatial MHSA ($MHSA_S$) to model global spatial correlation between keypoints within an identical frame. Each attention head ($i = 1, ..., h$) can be represented as:

$$head_i = Softmax(\frac{Q_S^i (K_S^i)'}{\sqrt{d}})V_S^i \tag{3}$$

where $'$ indicates the matrix transpose. All the attention heads h are concatenated together to form the $MHSA_S$:

$$MHSA_S(Q_S, K_S, V_S) = \textbf{cat}(head_1, ..., head_h)W_S \tag{4}$$

where $W_S \in \mathbb{R}^{d_m \times d_m}$ is the linear transformation matrix. Concurrently, H_{TN} as a residual adds the output of $MHSA_S$ to form the new output $H_S \in \mathbb{R}^{T \times N \times d_m}$, which is then fed into the layer normalization (LN) and MLP followed by a residual connection and LN. The formula can be represented as:

$$H_S = MHSA_S(Q_S, K_S, V_S) + H_{TN} \tag{5}$$

$$P_{NT} = MLP(LN(H_S)) + H_S \tag{6}$$

where $P_{NT} \in \mathbb{R}^{N \times T \times d_m}$ is the output of the Kinematics-Enhanced Transformer after being reshaped.

3.2. Trajectory-Enhanced Transformer

Trajectory-Enhanced Transformer aims to integrate the prior trajectory information of joint motion across frames into a sequence of tokens $P_{NT} \in \mathbb{R}^{N\times T\times d_m}$, in which each joint is regarded as an individual token in the time dimension. TPA first connects the identical keypoints (including itself) across neighboring frames to construct the temporal local topology, as shown in Figure 1, which is denoted as the symmetric affinity matrix $A_T \in \mathbb{R}^{T\times T}$. In order to enhance the global attention of temporal coherence in the MHSA, we simulate a temporal global topology that considers the implicit temporal correlation among neighboring and non-neighboring frames. These keypoints belonging to the identical trajectory among neighboring and non-neighboring frames are connected by the learnable vector (dotted line) to form the simulated temporal global topology, as shown in Figure 1. This topology can be expressed in the form of a learnable matrix $\hat{A}_T \in \mathbb{R}^{T\times T}$. Thus, the equation of joint motion trajectory topology can be represented as:

$$A_R = \frac{(A_T + \hat{A}_T) + (A_T + \hat{A}_T)'}{2} \tag{7}$$

where $'$ denotes the matrix transpose, $A_R \in \mathbb{R}^{T\times T}$ is a learnable affinity matrix. Then, we transform P_{NT} to embeddings $\bar{P}_{NT} \in \mathbb{R}^{N\times T\times d_m}$ by the linear transformation. Also, we utilize a learnable weight matrix $M_T \in \mathbb{R}^{T\times d_m}$ to allow different keypoints for different prior knowledge learning. The formula of one TPA is represented as:

$$TPA(P_{NT}) := (M_T \odot \bar{P}_{NT})A_R \tag{8}$$

We stack two TPA blocks with a residual connection to obtain the temporal tokens $H_{NT} \in \mathbb{R}^{N\times T\times d_m}$ as follows:

$$H_{NT} = TPA(TPA(P_{NT})) + P_{NT} \tag{9}$$

The learnable temporal positional embedding is then added to H_{NT}. After that, the $H_{NT} \in \mathbb{R}^{N\times T\times d_m}$ is converted into queries $Q_T \in \mathbb{R}^{N\times T\times d_m}$, keys $K_T \in \mathbb{R}^{N\times T\times d_m}$ and values $V_T \in \mathbb{R}^{N\times T\times d_m}$ by the linear transformation. We use a temporal MHSA ($MHSA_T$) to model the global temporal correlation between joints across all frames as follows:

$$head_i = Softmax(\frac{Q_T^i(K_T^i)'}{\sqrt{d}})V_T^i \tag{10}$$

$$MHSA_T(Q_T, K_T, V_T) = \mathbf{cat}(head_1, ..., head_h)W_T \tag{11}$$

where $W_T \in \mathbb{R}^{d_m\times d_m}$ is the linear transformation matrix. Similar to $MHSA_S$, we can obtain the final output H_{ST}:

$$H_T = MHSA_T(Q_T, K_T, V_T) + H_{NT} \tag{12}$$

$$H_{ST} = MLP(LN(H_T)) + H_T \tag{13}$$

where $H_{ST} \in \mathbb{R}^{N\times T\times d_m}$ is the output of Trajectory-Enhanced Transformer.

3.3. Stacked Spatio-Temporal Encoders

After being reshaped, H_{ST} is fed into the stacked spatio-temporal encoders which consist of alternating spatial and temporal transformers. The number of stacks is L. The sequential features are reshaped according to the type of the MHSA before fed into the encoder (spatial or temporal).

3.4. Regression Head

We utilize the linear layer as a regression head to predict the 3D pose sequence $\hat{P}_{3D} \in \mathbb{R}^{T\times N\times 3}$. The overall loss function for our network is given as:

$$\mathcal{L} = \mathcal{L}_W + \lambda_T\mathcal{L}_T + \lambda_M\mathcal{L}_M \tag{14}$$

where $\mathcal{L}_W$ denotes the weighted mean per-joint position error (WMPJPE) loss [42], $\mathcal{L}_T$ is the temporal consistency loss [10], and $\mathcal{L}_M$ indicates the mean per joint velocity error (MPJVE) loss [29]. Here λ_T and λ_M are hyper-parameters.

4. Experiments

4.1. Datasets and Protocols

Datasets. We evaluated our model on three public datasets, namely Human3.6M [12], MPI-INF-3DHP [24] and HumanEva [32]. Human3.6M is an indoor scenes dataset with 3.6 million video frames. It has 11 professional actors, performing 15 actions under 4 synchronized camera views. Following previous work [34, 42], we used subjects 1, 5, 6, 7 and 8 for training, and subjects 9 and 11 for testing. MPI-INF-3DHP is also a public large-scale dataset. Following the setting of [34, 42], we used the area under the curve (AUC), percentage of correct keypoints (PCK) and MPJPE as evaluation metrics. HumanEva is a smaller dataset. To have a fair comparison with [42, 46], we evaluated our method for actions (Walk and Jog) of subjects S1, S2, S3.

Protocol. Protocol#1 is denoted as the mean per-joint position error (MPJPE), which is the average euclidean distance in millimeters (mm) between the predicted and the ground-truth 3D joint coordinates. Protocol#2 refers to the reconstruction error after the predicted 3D pose is aligned to the ground-truth 3D pose using procrustes analysis [8], denoted as P-MPJPE (mm).

Implementation Details. We implemented our method in the Pytorch framework on one GeForce RTX 3090 GPU. The input 2D keypoints were detected by 2D pose detector [3] or 2D ground truth. The W in WMPJPE follows the setting of MixSTE [42]. We set the number of stacked spatio-temporal encoders L to 7. Thus, the encoders contain 14 spatial and temporal transformer layers [36] with number of heads h=8, feature size C=512. During the training stage, we use the Adam optimizer to train our model with a batch size of 7. The learning rate is initialized to 0.00007 and decayed by 0.99 per epoch. Recently, diffusion models have been introduced in 3D pose estimation [7, 31] and

Table 1. Quantitative comparison results with the state-of-the-art methods on Human3.6M. The 2D poses obtained by CPN [3] are used as inputs. **Top table**: evaluation results of MPJPE (mm); **Bottom table**: evaluation results of P-MPJPE (mm);

MPJPE (CPN)	**Publication**	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
UGCN [37] (T=96)†	ECCV'20	40.2	42.5	42.6	41.1	46.7	56.7	41.4	42.3	56.2	60.4	46.3	42.2	46.2	31.7	31.0	44.5
PoseFormer [46] (T=81)†	ICCV'21	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
StridedFormer [17] (T=351)†	TMM'22	40.3	43.3	40.2	42.3	45.6	52.3	41.8	40.5	55.9	60.6	44.2	43.0	44.2	30.0	30.2	43.7
GraFormer [45]	CVPR'22	45.2	50.8	48.0	50.0	54.9	65.0	48.2	47.1	60.2	70.0	51.6	48.7	54.1	39.7	43.1	51.8
MHFormer [18] (T=351)†	CVPR'22	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 [30] (T=243)†	ECCV'22	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 [42] (T=81)†	CVPR'22	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
MixSTE [42] (T=243)†	CVPR'22	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
DUE [41] (T=300)†	MM'22	37.9	41.9	36.8	39.5	40.8	49.2	40.1	40.7	47.9	53.3	40.2	41.1	40.3	30.8	28.6	40.6
GLA-GCN [40] (T=243)†	ICCV'23	41.3	44.3	40.8	41.8	45.9	54.1	42.1	41.5	57.8	62.9	45.0	42.8	45.9	29.4	29.9	44.4
POT [14]	AAAI'23	47.9	50.0	47.1	51.3	51.2	59.5	48.7	46.9	56.0	61.9	51.1	48.9	54.3	40.0	42.9	50.5
STCFormer [34] (T=81)†	CVPR'23	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
STCFormer [34] (T=243)†	CVPR'23	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
DiffPose [7] (T=243)†*	CVPR'23	33.2	36.6	33.0	35.6	37.6	45.1	35.7	35.5	46.4	49.9	37.3	35.6	36.5	24.4	24.1	36.9
D3DP [31] (T=243)†*	ICCV'23	33.0	34.8	31.7	33.1	37.5	43.7	34.8	33.6	45.7	47.8	37.0	35.0	35.0	24.3	24.1	35.4
Ours (T=81)†		39.1	41.9	37.3	40.1	44.0	51.3	39.8	41.0	51.4	56.0	43.0	41.0	42.6	28.8	29.5	41.8
Ours (T=243)†		37.3	39.2	35.9	37.6	42.5	48.2	38.6	39.0	51.4	55.9	41.6	39.0	40.0	27.0	27.4	40.1
Ours (T=243)†*		30.1	32.1	29.1	30.6	35.4	39.3	32.8	30.9	43.1	45.5	34.7	33.2	32.7	22.1	23.0	33.0

P-MPJPE (CPN)	**Publication**	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
UGCN [37] (T=96)†	ECCV'20	31.8	34.3	35.4	33.5	35.4	41.7	31.1	31.6	44.4	49.0	36.4	32.2	35.0	24.9	23.0	34.5
PoseFormer [46] (T=81)†	ICCV'21	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
StridedFormer [17] (T=351)†	TMM'22	32.7	35.5	32.5	35.4	35.9	41.6	33.0	31.9	45.1	50.1	36.3	33.5	35.1	23.9	25.0	35.2
P-STMO [30] (T=243)†	ECCV'22	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 [42] (T=81)†	CVPR'22	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
MixSTE [42] (T=243)†	CVPR'22	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
DUE [41] (T=300)†	MM'22	30.3	34.6	29.6	31.7	31.6	38.9	31.8	31.9	39.2	42.8	32.1	32.6	31.4	25.1	23.8	32.5
GLA-GCN [40] (T=243)†	ICCV'23	32.4	35.3	32.6	34.2	35.0	42.1	32.1	31.9	45.5	49.5	36.1	32.4	35.6	23.5	24.7	34.8
STCFormer [34] (T=81)†	CVPR'23	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
STCFormer [34] (T=243)†	CVPR'23	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
D3DP [31] (T=243)†*	ICCV'23	27.5	29.4	26.6	27.7	29.2	34.3	27.5	26.2	37.3	39.0	30.3	27.7	28.2	19.6	20.3	28.7
Ours (T=81)†		30.6	33.4	30.1	31.9	33.7	38.2	30.6	30.7	40.9	44.8	34.4	30.5	32.7	22.3	24.0	32.6
Ours (T=243)†		30.1	32.3	29.6	30.8	32.3	37.3	30.0	30.2	41.0	45.3	33.6	29.9	31.4	21.5	22.6	31.9
Ours (T=243)†*		24.1	26.7	24.2	24.9	27.3	30.6	25.2	23.4	34.1	35.9	28.1	25.3	25.9	17.8	18.8	26.2

T is the number of input frames. (†) denotes using temporal information, and (*) indicates the diffusion-based methods. Red: Best results. Blue: Runner-up results.

have achieved significant improvements in performance, as the diffusion process can be viewed as an augmentation method for pose data. To demonstrate the adaptability of our method, we introduced a diffusion process to our network, following the setting of D3DP [31], which also uses the transformer-based network as the backbone. We used KTPFormer as the denoiser in the D3DP [31]. For the design of the remaining diffusion process, our experimental parameters were set to be the same as D3DP [31].

4.2. Comparison with State-of-the-art Methods

Results based on Human3.6M. We compared our results with those of recent state-of-the-art methods based on the dataset Human3.6M. As shown in Table 1, our method (diffusion-based) achieves the state-of-the-art (SOTA) result 33.0mm in MPJPE and 26.2mm in P-MPJPE using the 2D poses detected by CPN [3] as inputs. Our method (diffusion-based) outperforms D3DP [31] by 2.4mm under MPJPE and 2.5mm under P-MPJPE with the same settings (the number of frames, hypotheses, and iterations) as D3DP [31]. This demonstrates that our network can serve as an excellent backbone for diffusion-based methods, effectively improving the performance for 3D pose estimation. Besides, we obtain the best results 40.1mm under T=243 setting and 41.8mm under T=81 setting in MPJPE among all methods that are not diffusion-based.

Table 2 compares our results with those of SOTA models using ground-truth 2D poses as inputs. Our method (diffusion-based) achieves the SOTA result 18.1mm, with the same settings (the number of frames, hypotheses, and iterations) as D3DP [31]. On the other hand, we obtain the best result 19.0mm under T=243 setting and 22.2mm under T=81 setting in MPJPE without diffusion process. Compared to GLA-GCN [40], there is a noticeable improvement (21.0→19.0mm) with T=243. Under T=81 setting, our method outperforms the second-best result by 3.5mm.

Results based on MPI-INF-3DHP. We evaluated the performance on MPI-INF-3DHP dataset to verify the generalization capability of our method. Following previous work [34, 44], we trained our model with ground-truth 2D poses as inputs. Table 3 shows the comparison results on the MPI-INF-3DHP test set. Our method with T=81 achieves the to-date best result with PCK of 98.9%, AUC of 85.9% and MPJPE of 16.7mm, outperforming the existing SOTA models by 0.2% in PCK, 2.0% in AUC and 6.4mm in MPJPE. Moreover, our method with T=27 also surpasses all other methods in all three metrics. These results demonstrate the strong generalization capability of our method.

Results based on HumanEva. Table 4 shows the performance comparison between ours and other methods on HumanEva dataset. Our method yields the best result of 15.3mm under T=27. Also, our method is superior to other algorithms under T=81. Due to the short video length in HumanEva, our method gives better results under T=27 than T=81. These results highlight the effectiveness of our method on small datasets.

Table 2. Quantitative comparison results of MPJPE (mm) with the state-of-the-art methods on Human3.6M using ground-truth (GT) 2D poses as inputs.

MPJPE (GT)	**Publication**	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
UGCN [37] (T=96)†	ECCV'20	23.0	25.7	22.8	22.6	24.1	30.6	24.9	24.5	31.1	35.0	25.6	24.3	25.1	19.8	18.4	25.6
PoseGTAC [47]	IJCAI'21	37.2	42.2	32.6	38.6	38.0	44.0	40.7	35.2	41.0	45.5	38.2	39.5	38.2	29.8	33.0	38.2
PoseFormer [46] (T=81)†	ICCV'21	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
StridedFormer [17] (T=351)†	TMM'22	27.1	29.4	26.5	27.1	28.6	33.0	30.7	26.8	38.2	34.7	29.1	29.8	26.8	19.1	19.8	28.5
GraFormer [45]	CVPR'22	32.0	38.0	30.4	34.4	34.7	43.3	35.2	31.4	38.0	46.2	34.2	35.7	36.1	27.4	30.6	35.2
MHFormer [18] (T=351)†	CVPR'22	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 [30] (T=243)†	ECCV'22	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
DUE [41] (T=300)†	MM'22	22.1	23.1	20.1	22.7	21.3	24.1	23.6	21.6	26.3	24.8	21.7	21.4	21.8	16.7	18.7	22.0
MixSTE [42] (T=81)†	CVPR'22	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
MixSTE [42] (T=243)†	CVPR'22	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
POT [14]	AAAI'23	32.9	38.3	28.3	33.8	34.9	38.7	37.2	30.7	34.5	39.7	33.9	34.7	34.3	26.1	28.9	33.8
STCFormer [34] (T=81)†	CVPR'23	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
STCFormer [34] (T=243)†	CVPR'23	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
GLA-GCN [40] (T=243)†	ICCV'23	20.1	21.2	20.0	19.6	21.5	26.7	23.3	19.8	27.0	29.4	20.8	20.1	19.2	12.8	13.8	21.0
DiffPose [7] (T=243)†*	CVPR'23	18.6	19.3	18.0	18.4	18.3	21.5	21.5	19.1	23.6	22.3	18.6	18.8	18.3	12.8	13.9	18.9
D3DP [31] (T=243)†*	ICCV'23	18.7	18.2	18.4	17.8	18.6	20.9	20.2	17.7	23.8	21.8	18.5	17.4	17.4	13.1	13.6	18.4
Ours (T=81)†		22.5	22.4	21.3	21.4	21.2	25.5	24.2	22.4	24.4	27.5	22.7	21.4	21.7	16.3	17.3	22.2
Ours (T=243)†		19.6	18.6	18.5	18.1	18.7	22.1	20.8	18.3	22.8	22.4	18.8	18.1	18.4	13.9	15.2	19.0
Ours (T=243)†*		18.8	17.4	18.1	17.7	18.3	20.6	19.6	17.7	23.3	22.0	18.7	17.0	16.8	12.4	13.5	18.1

T is the number of input frames. (†) denotes using temporal information, and (*) indicates the diffusion-based methods. Red: Best results. Blue: Runner-up results.

Table 3. Performance comparisons on MPI-INF-3DHP

Method	**Publication**	PCK↑	AUC↑	MPJPE↓
UGCN [37] (T=96)	ECCV'20	86.9	62.1	68.1
PoseFormer [46] (T=9)	ICCV'21	88.6	56.4	77.1
MHFormer [18] (T=9)	CVPR'22	93.8	63.3	58.0
MixSTE [42] (T=27)	CVPR'22	94.4	66.5	54.9
P-STMO [30] (T=81)	ECCV'22	97.9	75.8	32.2
Diffpose [7] (T=81)	CVPR'23	98.0	75.9	29.1
D3DP [31] (T=243)	ICCV'23	98.0	79.1	28.1
PoseFormerV2 [44] (T=81)	CVPR'23	97.9	78.8	27.8
GLA-GCN [40] (T=81)	ICCV'23	98.5	79.1	27.7
STCFormer [34] (T=27)	CVPR'23	98.4	83.4	24.2
STCFormer [34] (T=81)	CVPR'23	98.7	83.9	23.1
Ours (T=27)		98.9	84.4	19.2
Ours (T=81)		98.9	85.9	16.7

Metric ↑ denotes the higher, the better, ↓ denotes the lower, the better.

Table 4. The MPJPE evaluation results on HumanEva testset.

Method	Walk			Jog			**Avg**
	S1	S2	S3	S1	S2	S3	
TCN [29] (T=81)	13.1	10.1	39.8	20.7	13.9	15.6	18.9
PoseFormer [46] (T=43)	16.3	11.0	47.1	25.0	15.2	15.1	21.6
MixSTE [42] (T=43)	20.3	22.4	34.8	27.3	32.1	34.3	28.5
Ours (T=43)	16.5	13.9	19.9	25.3	15.9	16.5	18.0
Ours (T=27)	12.3	11.5	19.5	20.9	13.1	14.5	15.3

Table 5. Results of ablation study of each module in our KTPFormer on Human3.6M dataset.

Method	MPJPE (mm)	Parameters (M)	FLOPs (M)
Baseline	21.8	33.6506	139038
+KPA(w/o prior)	21.5	33.6501	139042
+KPA(w/o global)	20.7	33.6501	139042
+KPA	20.0	33.6501	139042
+TPA(w/o prior)	21.4	33.6527	139055
+TPA(w/o global)	20.5	33.6527	139055
+TPA	19.7	33.6527	139055
+KPA(w/o prior)+TPA(w/o prior)	21.4	33.6522	139059
+KPA(w/o global)+TPA(w/o global)	20.0	33.6522	139059
+KPA+TPA	19.0	33.6522	139059

Table 6. Results of ablation study involving different combinations of KPA and TPA in the network.

Method	MPJPE (mm)	Parameters (M)	FLOPs (M)
Baseline	21.8	33.6506	139038
United Mode (UMD)	20.0	33.6522	139059
Parallel Mode (PMD)	19.8	33.6512	139051
Separate Mode-S (SMD-S)	20.4	33.6512	139051
Separate Mode (SMD)	19.0	33.6522	139059

4.3. Ablation Study

Effect of each module To verify the effectiveness of the proposed modules, we conducted ablation experiments on Human3.6M (T=243) using ground-truth 2D poses as inputs. Table 5 presents the results of ablation study of each module. Our baseline network utilizes a linear layer to lift the 2D pose sequence to the high-dimensional space and then exploits the stacked spatio-temporal encoders (L=8) to predict the 3D pose sequence. As shown, the incorporation of KPA and TPA brings 1.8mm and 2.1mm of MPJPE drops, respectively. The prior knowledge of KPA and TPA contribute 1.5mm and 1.7mm of error drop, respectively. The global topologies of KPA and TPA yield reduction in MPJPE of 0.7mm and 0.8mm, respectively. With both KPA and TPA modules, the performance has improved 2.8mm. More remarkably, the number of parameters and FLOPs merely increase by 0.0016M and 21M, respectively, showing that our method is both effective and efficient.

Effect on different combinations of KPA and TPA. We analyzed the impacts to performance for four different combinations of KPA and TPA, including the United Mode (UMD), the Separate Mode (SMD), the Separate Mode-S and the Parallel Mode (PMD). UMD indicates that the output of the KPA is fed into the two TPA blocks with a residual connection, followed by stacked spatio-temporal encoders. SMD represents that KPA is followed by spatial MHSA and two TPA blocks with a residual connection are followed by temporal MHSA. The SMD-S differs from the SMD in that only one TPA block is followed by temporal MHSA. For PMD, the input is fed into TPA and Kinematics-Enhanced Transformer simultaneously, and the outputs of them are then added together and fed into temporal MHSA. Table 6 shows the comparison results of Human3.6M with T=243 frames between the four modes. The MPJPE result of UMD is worse than that of SMD because the features from KPA are fed directly into TPA, which

Table 7. Comparative results for various 3D pose estimators trained with and without KPA and TPA on Human3.6M dataset.

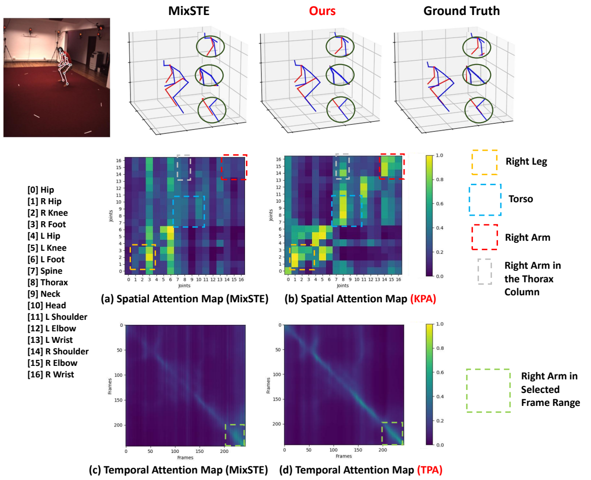
Method	MPJPE (mm)	Parameters (M)	FLOPs (M)
PoseFormer [46] (T=81)	31.3	9.558	815.522
+KPA+TPA	28.8 (-2.5)	9.560 (+0.02)	815.885 (+0.363)
StridedTransformer [17] (T=351)	28.5	3.979	801.093
+KPA+TPA	27.4 (-1.1)	3.980 (+0.01)	801.859 (+0.766)
MHFormer [18] (T=243)	30.9	24.767	4826.854
+KPA+TPA	29.1 (-1.8)	24.773 (+0.06)	4829.873 (+3.019)
MixSTE [42] (T=243)	21.6	33.650	139038.488
+KPA+TPA	19.0 (-2.6)	33.652 (+0.02)	139059.638 (+21.15)
STCFormer [34] (T=81)	25.7	4.747	6535.219
+KPA+TPA	25.1 (-0.6)	4.748 (+0.01)	6541.565 (+6.346)

leads to the confusion of spatial and temporal information. The comparison between PMD and SMD illustrates that TPA is more suitable to inject the trajectory information into high-dimensional tokens, rather than the initial 2D pose sequence. Besides, KPA and TPA should be independently followed by spatial and temporal MHSA, without interfering with each other. The comparison between the SMD-S and SMD indicates that the stacked TPA blocks can inject the prior information into pose tokens more effectively.

4.4. Qualitative Analysis and Discussion

We visualize in Figure 3 the 3D pose estimation results and attention maps so as to validate the efficacy of our method in comparison to MixSTE [42]. As shown, the spatial and temporal attention outputs from different heads are both averaged to show the distribution of attention weights of joints and frames. Figure 3(a) illustrates the phenomenon of attention collapse where the attention weights become highly concentrated on the right and left foot regions with other joints ($i.e.$, torso and right arm) being ignored in the spatial attention map of MixSTE [42], leading to poor predicted results of 3D pose (top of Figure 3). In contrast, the spatial attention weights (Figure 3(b)) are activated by KPA in regions of right arm, right leg and torso. In particular, the three joints of the right arm exhibit stronger attention weights in the thorax column, owing to the anatomical connection between the right hand and the torso. The attention allocation is, therefore, more reasonable (Figure 3(b)), contributing to an enhanced performance of 3D pose predicted by our method. Moreover, Figure 3(c) depicts the averaged temporal attention weights of the three joints of right arm. In contrast, TPA (Figure 3(d)) yields stronger temporal correlations along the *diagonal* as it connects the consecutive frames and small range of non-adjacent frames. In cases when input video records human motion of normal speed and the video has a high frame rate (e.g., 50 fps), the joints within a small range of frames would likely exhibit very small motion variations. Hence, such enhanced temporal attention can improve prediction accuracy.

Adaptable to Different 3D Pose Estimators. Our KPA and TPA are generic and can be applied in various transformer-based 3D pose estimators. To verify the adaptability, we selected five transformer-based 3D pose estimators as backbones. We removed the linear embedding before the first spatial encoder and put KPA in front of the first MHSA in these models. We used TPA to encode the features of different poses across frames on [17, 18, 46] and different joints across frames on [34, 42]. We trained these models on the Human3.6M dataset using 2D ground-truth poses as inputs. As shown in Table 7, our method brings about noticeable improvements in all the models in terms of MPJPE (mm), with very slight increases in the number of parameters and FLOPs, indicating that our KPA and TPA modules are lightweight and plug-and-play to different models for 3D pose estimation.



(a) Spatial Attention Map (MixSTE) (b) Spatial Attention Map (KPA)

(c) Temporal Attention Map (MixSTE) (d) Temporal Attention Map (TPA)

Figure 3. Comparison of visualization results and attention maps between ours and MixSTE [42]. The x-axis and y-axis correspond to the queries and the predicted outputs, respectively.

5. Conclusion

In this paper, we develop a Kinematics and Trajectory Prior Knowledge-Enhanced Transformer (KTPFormer), which explores two novel prior attention mechanisms (KPA and TPA) for 3D pose estimation. The KPA and TPA can enhance the capabilities of modeling global correlations in the self-attention mechanisms effectively. Experimental results on three benchmarks demonstrate that our method is effective in improving the performance with only a very small increase in the computational overhead. Moreover, our KPA and TPA can be integrated into various transformer-based 3D pose estimators as lightweight plug-and-play modules.

Acknowledgments

The work described in this paper is supported, in part, by the Innovation and Technology Commission of Hong Kong under grant ITP/028/21TP and by the Laboratory for Artificial Intelligence in Design (Project Code: RP1-1) under InnoHK Research Clusters, Hong Kong Special Administrative Region.

References

[1] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2272–2281, 2019. 3

[2] Tianlang Chen, Chen Fang, Xiaohui Shen, Yiheng Zhu, Zhili Chen, and Jiebo Luo. 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. 2

[3] 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. 2, 5, 6

[4] Yu Cheng, Bo Wang, Bo Yang, and Robby T Tan. Graph and temporal convolutional networks for 3d multi-person pose estimation in monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1157–1165, 2021. 3

[5] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2262–2271, 2019. 3

[6] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016. 3

[7] Jia Gong, Lin Geng Foo, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13041–13051, 2023. 2, 3, 5, 6, 7

[8] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40:33–51, 1975. 5

[9] Nate Hagbi, Oriel Bergig, Jihad El-Sana, and Mark Billinghurst. Shape recognition and pose estimation for mobile augmented reality. *IEEE transactions on visualization and computer graphics*, 17(10):1369–1379, 2010. 1

[10] 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)*, pages 68–84, 2018. 2, 5

[11] Wenbo Hu, Changgong Zhang, Fangneng Zhan, Lei Zhang, and Tien-Tsin Wong. Conditional directed graph convolution for 3d human pose estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 602–611, 2021. 3

[12] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 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. 5

[13] Branislav Kisacanin, Vladimir Pavlovic, and Thomas S Huang. *Real-time vision for human-computer interaction*. Springer Science & Business Media, 2005. 1

[14] Han Li, Bowen Shi, Wenrui Dai, Hongwei Zheng, Botao Wang, Yu Sun, Min Guo, Chenglin Li, Junni Zou, and Hongkai Xiong. Pose-oriented transformer with uncertainty-guided refinement for 2d-to-3d human pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1296–1304, 2023. 2, 3, 6, 7

[15] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 3

[16] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part II 12*, pages 332–347. Springer, 2015. 2

[17] Wenhao Li, Hong Liu, Runwei Ding, Mengyuan Liu, Pichao Wang, and Wenming Yang. Exploiting temporal contexts with strided transformer for 3d human pose estimation. *IEEE Transactions on Multimedia*, 25:1282–1293, 2022. 1, 6, 7, 8

[18] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13147–13156, 2022. 1, 2, 6, 7, 8

[19] Kenkun Liu, Rongqi Ding, Zhiming Zou, Le Wang, and Wei Tang. A comprehensive study of weight sharing in graph networks for 3d human pose estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 318–334. Springer, 2020. 3

[20] Mengyuan Liu and Junsong Yuan. Recognizing human actions as the evolution of pose estimation maps. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1159–1168, 2018. 1

[21] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017. 1

[22] Ruixu Liu, Ju Shen, He Wang, Chen Chen, Sen-ching Cheung, and Vijayan Asari. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5064–5073, 2020. 2

[23] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2640–2649, 2017. 2

[24] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *2017 international conference on 3D vision (3DV)*, pages 506–516. IEEE, 2017. 5

[25] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In *Proceed-*

ings of the IEEE/CVF international conference on computer vision, pages 10133–10142, 2019. 2

[26] Alejandro Newell, Kaiyu Yang, and Jia Deng. 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*, pages 483–499. Springer, 2016. 2

[27] Sungheon Park, Jihye Hwang, and Nojun Kwak. 3d human pose estimation using convolutional neural networks with 2d pose information. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 156–169. Springer, 2016. 2

[28] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7307–7316, 2018. 2

[29] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 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*, pages 7753–7762, 2019. 2, 5, 7

[30] Wenkang Shan, Zhenhua Liu, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. P-stmo: Pre-trained spatial temporal many-to-one model for 3d human pose estimation. In *European Conference on Computer Vision*, pages 461–478. Springer, 2022. 1, 6, 7

[31] Wenkang Shan, Zhenhua Liu, Xinfeng Zhang, Zhao Wang, Kai Han, Shanshe Wang, Siwei Ma, and Wen Gao. Diffusion-based 3d human pose estimation with multi-hypothesis aggregation. *arXiv preprint arXiv:2303.11579*, 2023. 2, 5, 6, 7

[32] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision*, 87(1-2):4–27, 2010. 5

[33] Mikael Svenstrup, Soren Tranberg, Hans Jorgen Andersen, and Thomas Bak. Pose estimation and adaptive robot behaviour for human-robot interaction. In *2009 IEEE International Conference on Robotics and Automation*, pages 3571–3576. IEEE, 2009. 1

[34] Zhenhua Tang, Zhaofan Qiu, Yanbin Hao, Richang Hong, and Ting Yao. 3d human pose estimation with spatio-temporal criss-cross attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4790–4799, 2023. 1, 2, 5, 6, 7, 8

[35] Bugra Tekin, Artem Rozantsev, Vincent Lepetit, and Pascal Fua. Direct prediction of 3d body poses from motion compensated sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 991–1000, 2016. 2

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 2, 5

[37] Jingbo Wang, Sijie Yan, Yuanjun Xiong, and Dahua Lin. Motion guided 3d pose estimation from videos. In *European Conference on Computer Vision*, pages 764–780. Springer, 2020. 3, 6, 7

[38] Tom Wehrbein, Marco Rudolph, Bodo Rosenhahn, and Bastian Wandt. Probabilistic monocular 3d human pose estimation with normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11199–11208, 2021. 2

[39] Tianhan Xu and Wataru Takano. Graph stacked hourglass networks for 3d human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16105–16114, 2021. 3

[40] Bruce XB Yu, Zhi Zhang, Yongxu Liu, Sheng-hua Zhong, Yan Liu, and Chang Wen Chen. 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*, pages 8818–8829, 2023. 3, 6, 7

[41] Jinlu Zhang, Yujin Chen, and Zhigang Tu. Uncertainty-aware 3d human pose estimation from monocular video. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5102–5113, 2022. 6, 7

[42] Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, and Junsong Yuan. 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*, pages 13232–13242, 2022. 1, 2, 5, 6, 7, 8

[43] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3425–3435, 2019. 3

[44] Qitao Zhao, Ce Zheng, Mengyuan Liu, Pichao Wang, and Chen Chen. 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*, pages 8877–8886, 2023. 1, 2, 6, 7

[45] Weixi Zhao, Weiqiang Wang, and Yunjie Tian. Graformer: Graph-oriented transformer for 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20438–20447, 2022. 2, 3, 6, 7

[46] 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*, pages 11656–11665, 2021. 1, 2, 5, 6, 7, 8

[47] Yiran Zhu, Xing Xu, Fumin Shen, Yanli Ji, Lianli Gao, and Heng Tao Shen. Posegtac: Graph transformer encoder-decoder with atrous convolution for 3d human pose estimation. In *IJCAI*, pages 1359–1365, 2021. 2, 3, 7

[48] Zhiming Zou and Wei Tang. Modulated graph convolutional network for 3d human pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11477–11487, 2021. 3