Human-Structure-Aware Token Position Embedding for Tokenized Pose Estimation

Zejun Gu, Zhong-Qiu Zhao, Henghui Ding, *Member, IEEE*, Hao Shen, Zhenhua Tang, Zhao Zhang, *Senior Member, IEEE*, and De-Shuang Huang, *Fellow, IEEE*

***Abstract*—Tokenized pose estimation (TPE) has demonstrated remarkable performance in lightweight human pose estimation (HPE) models. However, existing TPE methods typically initialize keypoint tokens randomly, without explicitly incorporating human structure priors. These priors play a vital role in HPE by effectively mitigating common challenges such as occlusion and ambiguity. To this end, we propose a Structure-Aware Keypoint Position Embedding (SAKPE). This embedding explicitly encodes inherent structural properties of the human body, such as symmetry and order, into the positional coordinates of keypoint tokens. It also employs learnable scale and offset factors to adapt to diverse human poses, thereby fully exploiting the geometric constraints among keypoints. Furthermore, to better leverage the positional relationships among patch tokens, we introduce a Layer-adaptive Hybrid Patch Position Embedding (LHPPE). It dynamically fuses absolute and relative position embeddings of patch tokens based on attention distributions across Transformer layers, enabling the model to learn both absolute and relative positional information adaptively. Taking the two together, we propose a novel position embedding method for pose estimation, named Human-structure-aware Token Position Embedding (HTPE). It significantly improves the performance of various TPE models. Extensive experiments on COCO, CrowdPose, and OCHuman show that HTPE achieves state-of-the-art (SOTA) performance among lightweight methods, with a negligible increase in parameters and FLOPs. Notably, it demonstrates consistent improvements under occlusion,, achieving up to 3.3 AP gains. The source code can be found in https://github.com/guzejungithub/HTPE**

Received 3 April 2026; revised 27 May 2026; accepted 27 May 2026. Date of publication 11 June 2026; date of current version 17 June 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62472137, Grant 62502006, and Grant 62472104; in part by Anhui Special Support Plan for High-Level Talents; and in part by the Scientific Research Foundation for Highlevel Talents of Anhui University of Science and Technology under Grant 2025yjrc0015. The associate editor coordinating the review of this article and approving it for publication was Prof. Jochen Lang. *(Corresponding author: Zhong-Qiu Zhao.)*

Zejun Gu, Zhong-Qiu Zhao, and Zhao Zhang are with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China (e-mail: guzejunmail@gmail.com; z.zhao@hfut.edu.cn; cszzhang@gmail.com).

Henghui Ding is with the Institute of Big Data, Fudan University, Shanghai 200433, China (e-mail: henghui.ding@gmail.com).

Hao Shen is with the School of Public Security and Emergency Management, Anhui University of Science and Technology, Hefei 231131, China (e-mail: haoshenhs@gmail.com).

Zhenhua Tang is with the Department of Computer and Information Science, University of Macau, Macau, SAR, China (e-mail: zhenhuat@foxmail.com).

De-Shuang Huang is with Ningbo Institute of Digital Twin and Ningbo Key Laboratory of Multi-Omics and Multimodal Biomedical Data Mining and Computing, Eastern Institute of Technology, Ningbo 315201, China (e-mail: dshuang@eitech.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2026.3700936, provided by the authors.

Digital Object Identifier 10.1109/TIP.2026.3700936

***Index Terms*—Lightweight human pose estimation, human-structure-aware, hybrid patch position embedding.**

I. Introduction

2D HUMAN pose estimation is a fundamental task in computer vision that aims to locate anatomical keypoints of the human body in images [1], [2], [3]. It has attracted widespread attention from both academia and industry, as it supports downstream tasks such as 3D pose estimation, human action analysis, medical diagnosis, and virtual reality [4], [5], [6], [7].

In recent years, several studies [8], [9] have achieved impressive accuracy in HPE. However, these models suffer from high computational complexity, large storage requirements, and long processing times. As a result, they are difficult to deploy on edge computing devices and fail to enable real-time deployment on embodied-AI platforms.

Meanwhile, some other approaches [10], [11], [12], [13] offer significant advantages in terms of computational speed and storage efficiency, but they suffer from limited accuracy, which can negatively impact the performance of downstream tasks. Therefore, how to design models that are both lightweight and high-performing remains a challenging problem.

Tokenized pose estimation [14], [15], [16], [17] has made significant progress in improving the performance of lightweight models. TokenPose [14] proposed representing each keypoint as a token, integrating the learning of visual cues and constraint relationships within a unified framework. This approach significantly reduces the number of parameters and computational cost while maintaining competitive performance. PPT [15] further introduced a Human Token Identification (HTI) module to locate human regions and crop out background tokens, thereby reducing computational cost while preserving pose estimation accuracy.

DistilPose [16] presented a Token Distilling Module (TDE) and Simulated Heatmaps to maximize knowledge transfer from a heatmap-based teacher model to a regression-based student model, improving the performance of lightweight models while maintaining low computational cost. SDPose [17] designed a Multi-Cycled Transformer (MCT) module that circulates tokens within Transformer layers to enhance the model's latent depth without increasing parameters. It also proposed a self-distillation paradigm that extracts knowledge from the MCT and transfers it to a single-pass model, achieving a balance between performance and resource consumption.

However, these methods suffer from the following issues: ***(1) Random initialization of keypoint tokens:***Current approaches initialize keypoint tokens randomly without incorporating any prior knowledge, as shown in Figure 1. In HPE, structure priors are crucial, as they reflect inherent spatial relationships between keypoints. ***(2) Lack of relative positional information of patch tokens:*** Existing methods use only absolute position embedding to encode the position of each patch. However, they overlook relative positional information between patches, which is essential for modeling the positional relationship among different body parts. ***(3) Uniform position embedding across layers:*** As shown in Figure 6, different Transformer layers focus on different regions. From the first to the last layer, the model's attention gradually converges from global to local. So, the importance of relative and absolute positional information varies across layers.

However, current methods use a uniform position embedding for all layers, lacking layer-specific position encoding tailored to the characteristics of each Transformer layer. To address the above issues, we propose a novel token position embedding method, Human-structure-aware Token Position Embedding (HTPE). As illustrated in Figure 1, HTPE is designed to fully capture the positional relationships between tokens while incorporating prior knowledge of human body structure, and it adaptively adjusts to the characteristics of different Transformer layers. HTPE consists of two main components: a Structure-Aware Keypoint Position Embedding module (SAKPE) and a Layer-adaptive Hybrid Patch Position Embedding (LHPPE) module.

First, we present the Structure-Aware Keypoint Position Embedding (SAKPE), which explicitly encodes the inherent structural properties of the human body into token position coordinates. These properties include symmetry (*e.g.,* left and right shoulders), order (*e.g.,* the sequence of shoulder, hip, knee, and ankle), and rigid topological structure (*e.g.,* fixed connectivity between keypoints) (see Figure 2). In real-world scenarios, human poses often deviate from these structural norms and can vary significantly in scale. To address this, we introduce learnable scale and offset factors to adapt to different poses. Furthermore, SAKPE employs Keypoint-based Rotary Position Embedding (K-RoPE) to capture the relative spatial relationships among keypoints. To the best of our knowledge, this is the first work in HPE to incorporate such structural properties into token position embeddings.

In addition, we propose the Layer-adaptive Hybrid Patch Position Embedding that combines both absolute and relative positional information. It not only provides absolute location information for each patch token but also uses rotary position embedding to capture relative spatial relationships between patch tokens, which is crucial for modeling body part interactions. Moreover, this module adaptively adjusts the weights of absolute and relative position embedding based on the attention distribution of different Transformer layers, enabling more effective learning of patch positions. HTPE achieves state-of-the-art performance for lightweight models, with negligible increases in FLOPs and parameters, and only a minor increase in inference latency and memory consumption. As shown in Figure 3, our method delivers significant performance improvements across models of various sizes. Among all tokenized pose estimation approaches, our method stands out as the most efficient and best-performing.

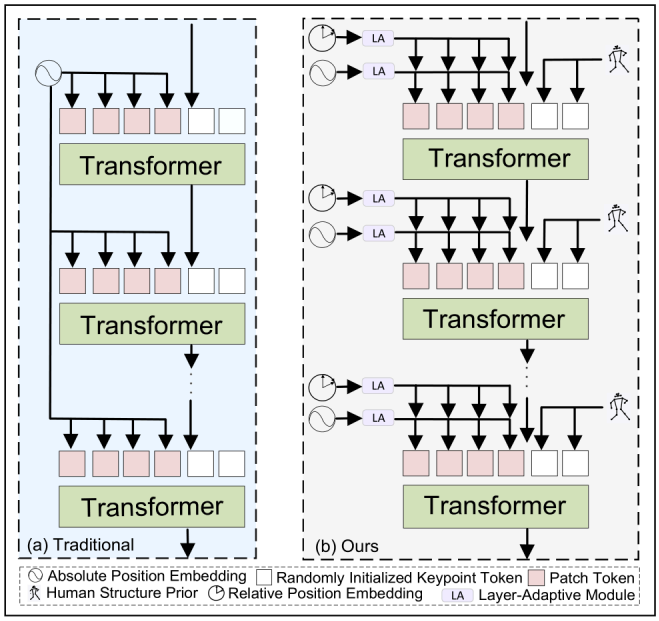

Fig. 1. Conceptual illustration of (a) traditional tokenized pose estimation methods and (b) our proposed method. Traditional methods randomly initialize keypoint tokens and use only absolute position embedding to provide location information for patch tokens. In contrast, our method explicitly encodes human structural priors into keypoint position embedding and adopts layer-adaptive hybrid patch position embedding to provide richer spatial information for patch tokens.

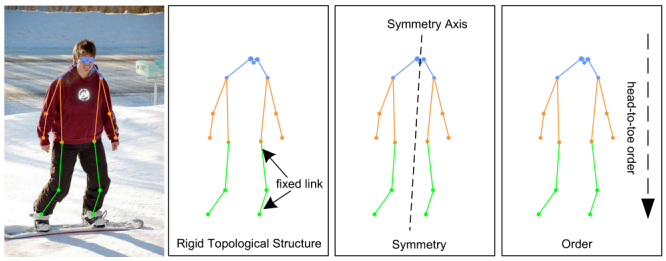

Fig. 2. An illustrative example showing the inherent structural properties of the human body. Regardless of variations in human pose, the anatomical keypoints consistently preserve a rigid topological structure, together with a certain degree of symmetry and order.

Our contributions are summarized as follows:

- We are the first to encode human structural characteristics and apply them to keypoint token position embedding. Combined with learnable scale and offset factors, this approach explicitly leverages prior knowledge of keypoint spatial relationships.
- We design a novel layer-adaptive hybrid patch position embedding, which not only learns both absolute position information and relative spatial relationships of patch tokens, but also adaptively adjusts their weight based on the characteristics of different Transformer layers, enabling more effective utilization of positional information of patch tokens.
- Experiments on the COCO [18], CrowdPose [19], and OCHuman [20] datasets demonstrate that our method achieves SOTA performance with nearly unchanged

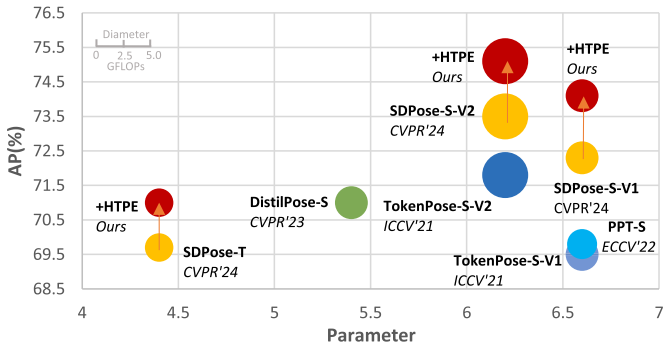

Fig. 3. Comparisons between other lightweight models and our proposed HTPE on the COCO validation set. HTPE achieves **SOTA** performance with a negligible increase in computation (GFLOPs) and parameters.

FLOPs and parameters, and a slight increase in inference latency and memory consumption. Meanwhile, it serves as an effective solution for occlusion. Moreover, our method is generalizable and can be applied into various tokenized pose estimation frameworks and other structured vision tasks, consistently delivering significant performance gains.

II. Related Work

A. Human Pose Estimation

1) Human Pose Estimation Based on Keypoint Modeling: Early research [21], [22] on human pose estimation mainly focused on keypoint modeling. Heatmap-based methods [14], [17], [23], [24] represent the positions of joints using probability heatmaps, while regression-based methods [10], [11], [12] model keypoints through coordinate vectors. Since the pioneering work proposed by Tompson et al. [25], heatmap-based pose estimation has dominated the field in terms of performance. Many subsequent works [9], [26] have designed powerful convolutional neural network (CNN) models for heatmap estimation in single-person pose estimation. Some efforts [24], [27] improved pose estimation accuracy by reducing quantization errors caused by heatmaps. However, heatmap-based methods often suffer from low efficiency and slow speed. Early regression-based studies [21], [22] proposed directly regressing joint coordinates from images. CenterNet [28] realized multi-person pose estimation within a single-stage object detection framework by directly regressing keypoint coordinates instead of bounding boxes. RLE [10] improved regression learning frameworks by quantifying uncertainty in coordinate regression. Although regression-based methods are more efficient, their accuracy generally lags behind heatmap-based approaches. Moreover, neither heatmap-based nor regression-based methods explicitly model the dependencies among keypoints.

2) Human Pose Estimation Based on Human Structure Priors: Some studies [29], [30] have proposed using deformable models based on anatomical priors to capture relationships between human joints. Chu et al. [31] introduced geometrical transform kernels to fuse features from different channels, which are assumed to correspond to different joints. Chen et al. [32] designed a pose discriminator to eliminate unreasonable pose estimations and encourage the predictor to learn structurally plausible poses. Geng et al. [1] represented human poses as discrete substructure tokens composed of multiple interdependent joints, predicted the categories of these tokens using a classifier, and then reconstructed the full pose with a pretrained decoder. Raychaudhuri et al. [33] proposed to explicitly incorporate anatomical priors by using a manifold of plausible human 2D poses, which effectively improves structural plausibility and cross-domain generalization. Yoo and Russakovsky [34] proposed the BPLP-C metric and constrained the transformation matrix to improve the consistency of body part length proportions, thereby further enhancing pose estimation performance. Wang et al. [35] presented Pose Prior Learner (PPL), which stores prototypical poses in a hierarchical memory and distills them into a general pose prior applicable to any object category, effectively mitigating the challenge of unsupervised categorical prior learning in pose estimation. Peng et al. [36] proposed KTPFormer, which introduces Kinematics Prior Attention (KPA) and Trajectory Prior Attention (TPA). These mechanisms incorporate anatomical structure and motion trajectory information into the self-attention process, overcoming the limitation of traditional Transformers, where Q, K, and V vectors are derived solely from simple linear mappings. Han et al. [37] introduced a method that models skeletons as graph structures and leverages graph neural networks to detect skeletal anomalies, thereby generating high-quality pseudo-labels, which significantly improves cross-domain animal pose estimation accuracy. ***In contrast, we propose Human-structure-aware Token Position Embedding (HTPE), specifically designed for lightweight Transformer-based tokenized human pose estimation (HPE). The proposed HTPE method encodes anatomical priors directly into the positional information of keypoint tokens in a general and computationally efficient manner, effectively enhancing the model's performance.***

3) Transformer-Based Human Pose Estimation: In recent years, the introduction of Transformers has driven significant progress in human pose estimation. TFPose [38] was the first to incorporate Transformers into a pose estimation framework using a regression approach. PRTR [12] proposed a two-stage end-to-end regression framework based on cascaded Transformers, achieving state-of-the-art performance among regression methods. TransPose [39] applied Transformers to heatmap-based human pose estimation, attaining competitive results. ViTPose [8] used a pure, non-hierarchical vision Transformer as the backbone network to extract features of single-person instances, demonstrating excellent performance, scalability, and flexibility. Poseur [11] employed a deformable cross-attention mechanism to extract keypoint features.

4) Lightweight Human Pose Estimation: TokenPose [14] input keypoint and visual tokens into the Transformer layers for feature extraction, then used keypoint tokens to predict heatmaps. DistilPose [15] designed a novel simulated heatmap loss to enable knowledge transfer from a heatmap-based teacher model to a regression-based student model. PPT [16] identified and discarded background tokens at different Transformer layers to reduce computational complexity. SDPose [17] introduced a self-distillation method with multi-cycle Transformers to alleviate underfitting and improve model

performance without increasing the number of Transformer layers. However, all these methods randomly initialize the keypoint tokens without any prior knowledge and rely solely on absolute position embedding to represent the position of patch tokens. They not only lack explicit utilization of human structural priors but also fail to learn the spatial relationships between patches fully.

B. Token Position Embedding

The Transformer was proposed in [42] and has achieved great success. It employs absolute positional encoding to capture token positions. To further leverage relative positional information, Shaw et al. [43] proposed relative positional bias (RPB). iRPE [44] improved upon RPB by multiplying relative position encoding with the query vectors. LaPE [45] introduced independent layernorm layers for each Transformer layer to optimize position encoding. RoFormer [46] was the first to propose Rotary Position Embedding (RoPE) and applied it in natural language processing. Mixed RoPE [47] extended RoPE to two-dimensional vision tasks, effectively capturing the complex spatial relationships between tokens. However, these methods have not yet been applied to human pose estimation and lack adaptive adjustments tailored for pose estimation. To address the above issues, our proposed HTPE incorporates human structure awareness and adaptively learns token positional information based on the attention distributions of different Transformer layers. It improves the performance of tokenized pose estimation with a small overall computational overhead.

III. Methods

This section provides a detailed description of our proposed Human-structure-aware Token Position Embedding (HTPE). The overall framework is illustrated in Figure 5. HTPE mainly includes two modules: a Structure-Aware Keypoint Position Embedding (SAKPE) and a Layer-adaptive Hybrid Patch Position Embedding (LHPPE). SAKPE incorporates inherent human structural priors to learn constraint cues between keypoints. LHPPE adaptively learns absolute and relative positional information based on the characteristics of different Transformer layers.

A. Structure-Aware Keypoint Position Embedding

SAKPE first encodes the positional coordinates for each keypoint token according to human structural characteristics. It then learns the relative spatial relationships between keypoints using Keypoint-based Rotary Position Embedding (K-RoPE).

1) Keypoint Token Coordinate Encoding: We encode the positional coordinates for each keypoint token based on the inherent structural properties of the human body. These properties impose intrinsic relative spatial constraints among keypoints, which are generally maintained to some extent despite pose variations. As shown in Figure 4, the structure of human pose exhibits three key characteristics: rigid topological structure, symmetry, and order.

a) Rigid topological structure: Rigid topological structure refers to fixed topological relationships between human keypoints, rigid bones, and limited range of joint rotation and movement [40], [41], as illustrated in Figure 4(a). It ensures that the relative positional relationships among keypoints remain relatively stable across different poses. This structural stability provides reliable geometric constraints for human pose modeling, enhancing the generalization and robustness of structure-aware encoding across various scenarios. The physiological basis for the rigid topological structure is provided by the supplementary material.

b) Symmetry: As shown in Figure 4(b), human keypoints are symmetrically distributed relative to the body's midline. The nose lies on the axis of symmetry, while other keypoints appear in pairs on the left and right sides of the body—such as the left and right eyes, shoulders, and hips. Even when the pose changes, this symmetrical structure is largely preserved. We define the body's midline as the central axis and divide the body into left and right half-planes. We then encode the direction of each keypoint relative to the midline using positive and negative signs: keypoints on the left side are assigned negative values, those on the right are assigned positive values.

Symmetrical keypoint pairs are assigned the same absolute value. The sign encodes directional priors, while the absolute value identifies pairing relationships between keypoints, thereby embedding the symmetry prior of human structure into the token position encoding.

c) Order: As depicted in Figure 4(c), human keypoints exhibit clear positional order, which manifests in two dimensions: along the body's midline, *i.e.,* the vertical direction, and perpendicular to the midline, *i.e.,* the horizontal direction. For example, from top to bottom, the keypoints follow the order: eyes, nose, shoulders, hips, knees, and ankles; from left to right: left shoulder, left eye, nose, right eye, right shoulder.

To convert symmetry and order into encodable information usable by the model, we define a 2D coordinate system with the top of the head as the origin. In this system, the Y-axis represents the direction along the body's midline, while the X-axis represents the perpendicular direction. As illustrated in Figure 4(c), according to its relative spatial position, each keypoint is assigned a unique coordinate value (x_{kpt}, y_{kpt}). In addition to symmetry, this coordinate encoding further captures other relative spatial relationships between keypoints: (1) Adjacent positions: For example, the closeness of the coordinate values between the eyes and the nose reflects their spatial proximity. (2) Relative distance: For example, the coordinate distance between the eyes and the feet is greater than that between the eyes and the nose, which provides a prior for the relative distance relationships between different parts of the human body.

2) Learnable Scale and Offset Factors: In fact, due to variations in human body sizes and diverse pose changes, real-world poses may somewhat deviate from the theoretical structural characteristics. As shown in Figure 4(d), this person's left and right ankles do not satisfy symmetry. Besides that, fixed coordinates cannot accommodate various body sizes. To address this issue, we introduce a learnable scale factor ξ and an offset factor β. The learnable scale factor adjusts keypoint coordinates to account for variations in body

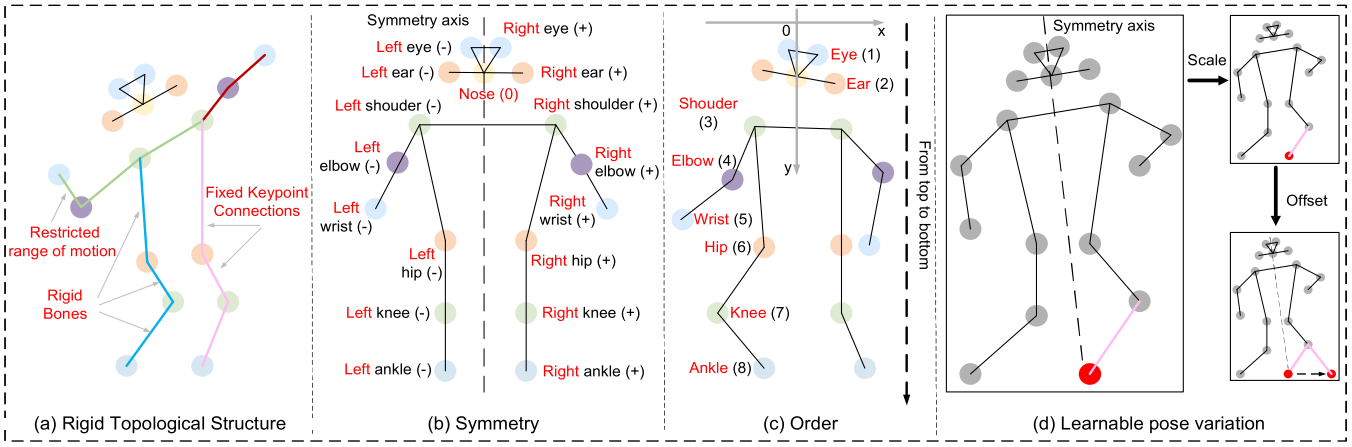

Fig. 4. Illustration of human structure characteristics (a b c) and our proposed learnable pose variation (d). (a) **Rigid topological structure.** Regardless of variations in human pose, the connections between anatomical keypoints remain fixed, the bones are rigid, and the range of limb movement is limited [40], [41]. (b) **Symmetry.** The left and right halves of the human body are symmetric about the central axis. (c) **Order.** The body maintains a relative order from head to foot, which also applies to the left-to-right direction. (d) **Learnable pose variation.** HTPE adapts to different poses by learnable scale and offset factors.

size, while the learnable offset captures deviations of keypoints from the idealized human structure. Specifically, we combine the learnable scale and offset factor with the keypoint token coordinates to form new position coordinates:

$$\tilde{x}^m_{kpt} = \xi^m_x x_{kpt} + \beta^m_x, \tag{1}$$

$$\tilde{y}^m_{kpt} = \xi^m_y y_{kpt} + \beta^m_y, \tag{2}$$

where $x_{kpt}, y_{kpt} \in \mathbb{R}^{N_{kpt}\times 1}$ represent the relative position coordinates of keypoints, $\boldsymbol{\xi}^m_x, \boldsymbol{\xi}^m_y \in \mathbb{R}^{1\times d/2}$ refer to the corresponding learnable scale, $\boldsymbol{\beta}^m_x, \boldsymbol{\beta}^m_y \in \mathbb{R}^{1\times d/2}$ represent the corresponding learnable offset, and $\tilde{x}^m_{kpt}, \tilde{y}^m_{kpt} \in \mathbb{R}^{N_{kpt}\times d/2}$ represents the adjusted coordinate. m is the index of the Transformer layer, d is the dimension of position embeddings, and N_{kpt} represents the number of keypoint tokens. The scale factor $\boldsymbol{\xi}^m_x$ is initialized to 1.0, and the offset factor $\boldsymbol{\beta}^m_x$ is initialized to 0.

3) Keypoint-Based Rotary Position Embedding: After keypoint token coordinate encoding, we seek to capture the ralative positional relationships between keypoints from these coordinates.

For this purpose, we propose a keypoint-based rotary position embedding (K-RoPE). While existing research on rotary position embedding focuses on patch tokens, we are the first to introduce the rotary position embedding specifically for keypoint tokens. Due to the rigid topological structure of the human body, the position information in horizontal and vertical directions is closely related. For example, when the arm rotates, both the x and y coordinates of the wrist change simultaneously with the arm's rotation angle. Previous studies [42], [46], [48] often treated the X-axis and Y-axis position information independently, which fails to effectively capture the interaction between the two directions. To address this, we introduce a learnable frequency ω to adaptively fuse X-axis and Y-axis information, enabling the model to fully capture the spatial relationships between horizontal and vertical directions:

$$\theta^m_{kpt} = \tilde{x}^m_{kpt}\boldsymbol{\omega}^m_{x_{kpt}} + \tilde{y}^m_{kpt}\boldsymbol{\omega}_{y^m_{kpt}}, \tag{3}$$

where $\theta^m_{kpt} \in \mathbb{R}^{N_{kpt}\times d/2}$ is the angle corresponding to the keypoint token of the m-th Transformer layer, and $\boldsymbol{\omega}^m_x, \boldsymbol{\omega}^m_y \in \mathbb{R}^{1\times d/2}$ refer to the corresponding learnable frequency.

To emphasize the role of the relative positional relationship of keypoints, we adopt rotary position embedding $e^{i\theta}$ to represent the position information of keypoint tokens. In addition, for a more direct expression of the positional relationships between keypoints, we apply rotary position embedding to the similarity computation between the query vector q_{kpt} and the key vector k_{kpt} (see Figure 5(c)). Specifically, we first convert $q_{kpt} \in \mathbb{R}^{N_{kpt}\times d}$ and $k_{kpt} \in \mathbb{R}^{N_{kpt}\times d}$ to complex-valued vectors $\bar{q}_{kpt} \in \mathbb{R}^{N_{kpt}\times d/2}$ and $\bar{k}_{kpt} \in \mathbb{R}^{N_{kpt}\times d/2}$, where the even-indexed elements are treated as the real part and the odd-indexed elements as the imaginary part of the complex numbers. This transformation allows the model to effectively capture the relative positional relationships of different body parts across multiple scales from a frequency perspective (see supplementary material for the motivation). Then, we apply a rotary transformation to them (see Figure 5(d)):

$$q'_{kpt} = \bar{q}_{kpt} e^{i\theta_{kpt}}, \tag{4}$$

$$k'_{kpt} = \bar{k}_{kpt} e^{i\theta_{kpt}}, \tag{5}$$

where $q'_{kpt} \in \mathbb{R}^{N_{kpt}\times d/2}$ is the rotated query vector, and $k'_{kpt} \in \mathbb{R}^{N_{kpt}\times d/2}$ is the rotated key vector. Finally, we replace the original q_{kpt} and k_{kpt} with q'_{kpt} and k'_{kpt} to compute the attention similarity.

B. Layer-Adaptive Hybrid Patch Position Embedding

After learning the human body structure priors among keypoint tokens through the SAKPE, we further capture the positional relationships between patch tokens using a Layer-adaptive Hybrid Patch Position Embedding (LHPPE). The keypoint position embedding learns relational constraints from high-level semantic priors, while the patch position embedding captures positional information from low-level visual cues. Existing tokenized pose estimation methods typically use absolute position embedding to encode the positional locations

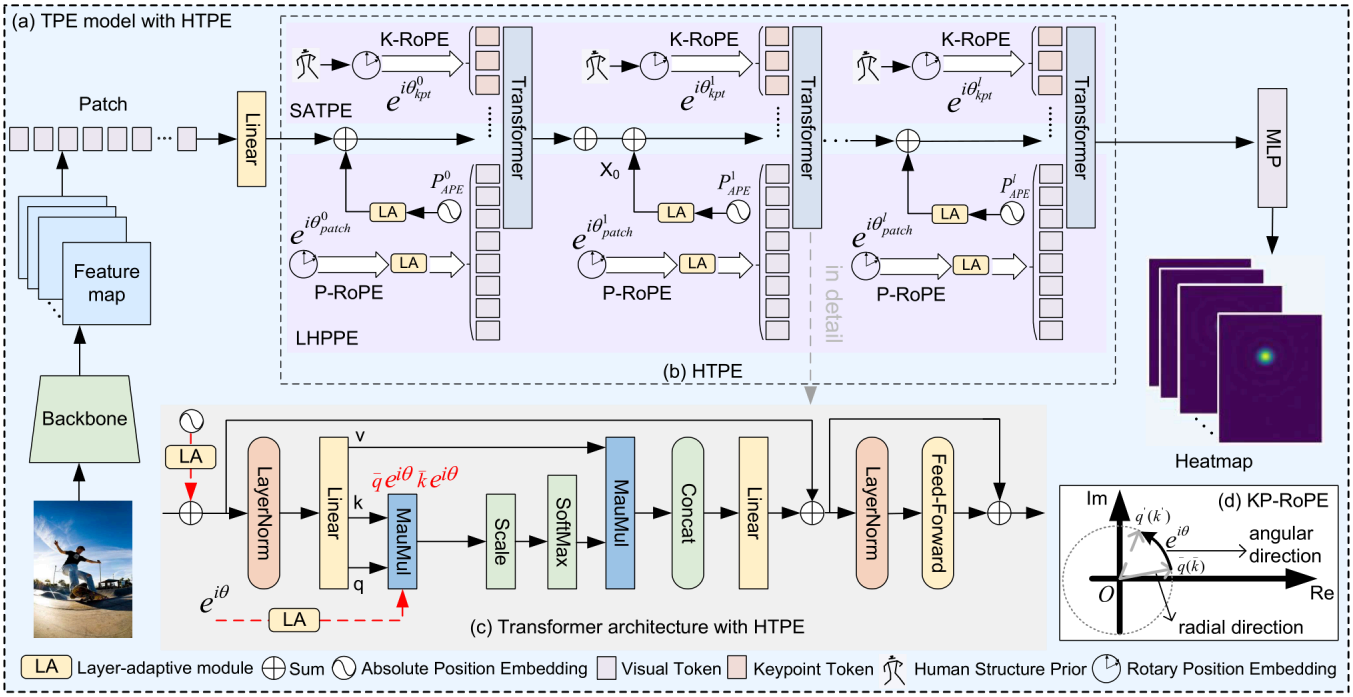

Fig. 5. **(a)** Overall architecture of the TPE model using HTPE. **(b)** Illustration of HTPE. **(c)** Details of the Transformer layer with HTPE. In this subfigure, the black feature boxes represent the main feature flow, while the red arrows emphasize the propagation path of HTPE. **(d)** KP-RoPE in the complex plane. Color legend: different colors indicate different modules and embeddings, as shown at the bottom.

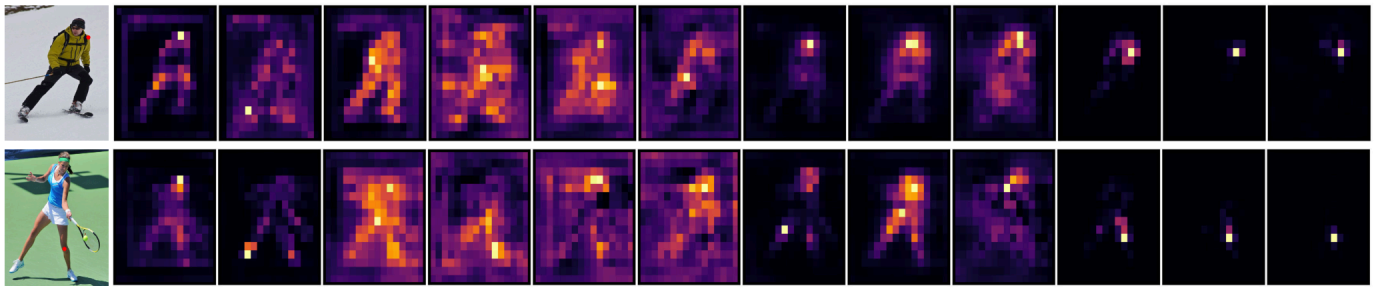

Fig. 6. Visualization of the attention maps between the keypoint token and visual tokens in different layers of SDPose-S-V1. From left to right are the 1st through 12th layers, respectively. **Top:** Left shoulder. **Bottom:** Left knee.

of patch tokens. However, for HPE, the relative positional relationships among body parts are also important. To address this, we propose a Keypoint-Patch coupled Rotary Positional Embedding (KP-RoPE) that supplements patch tokens with relative positional information. ***The KP-RoPE builds upon the K-RoPE by incorporating Patch-based Rotary Position Embedding (P-RoPE)***. It combines patch tokens and keypoint tokens to model the spatial relationships among various body parts jointly. Similar to the keypoint-based approach, we compute the rotated query and key representations for patches based on their position coordinates as follows:

$$q'_{patch} = \bar{q}_{patch} e^{i(x_{patch}\omega_{x_{patch}} + y_{patch}\omega_{y_{patch}})}, \tag{6}$$

$$k'_{patch} = \bar{k}_{patch} e^{i(x_{patch}\omega_{x_{patch}} + y_{patch}\omega_{y_{patch}})}, \tag{7}$$

where x_{patch}, $y_{patch} \in \mathbb{R}^{N_{patch}\times 1}$ represent the position coordinates of patches, and $\omega_{x_{patch}}, \omega_{x_{patch}} \in \mathbb{R}^{N_{patch}\times d/2}$ denotes the corresponding learnable frequency. $\bar{q}_{patch}, \bar{k}_{patch} \in \mathbb{R}^{N_{patch}\times d/2}$ represent the complex-valued query vector and key vector, $q'_{patch}, k'_{patch} \in \mathbb{R}^{N_{patch}\times d/2}$ refer to the rotated query vector and key vector, and N_{patch} is the number of patches.

Finally, the q'_{patch}, k'_{patch} together with the rotated query and key vectors of the keypoint tokens q'_{kpt}, k'_{kpt} are incorporated into the attention calculation:

$$A' = Re[(q')^{\top} k'], \tag{8}$$

where $q' = \{q'_{kpt}, q'_{patch}\}$, $k' = \{k'_{kpt}, k'_{patch}\}$, and Re$[\cdot]$ is the real part of the complex number.

The reason for adopting KP-RoPE is as follows: (1) Rotary position embedding maintains the relative independence between positional and semantic information of different human body parts, preventing positional cues from being overwhelmed and highlighting their importance. It is crucial for position-sensitive pose estimation. (2) Directly applying position embedding in the similarity computation clearly reflects the spatial relationships between patches, between keypoints, and between patches and keypoints, while keeping

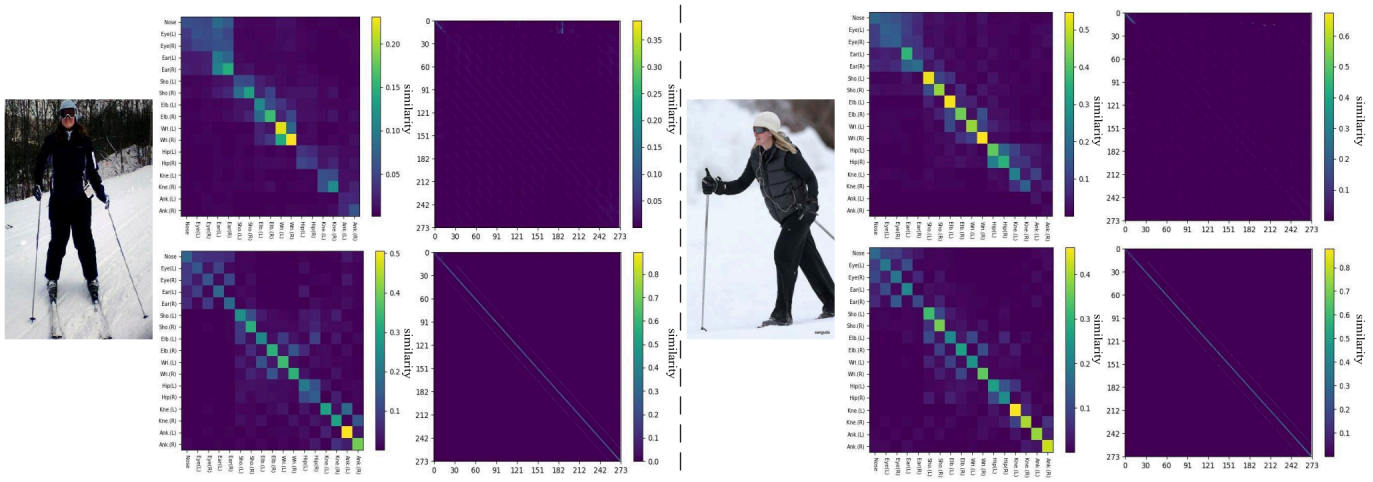

Fig. 7. Comparison of token attention maps of SDPose-S-V1 with HTPE **(bottom)** and without HTPE **(top)**.

semantic similarity and positional similarity decoupled. (3) The relationships between human keypoints naturally decay with increasing distance. Rotary position embedding effectively realizes this characteristic. The supplementary material provides a detailed analysis and proof of the above.

To explore the specific role of each Transformer layer, we visualized the attention maps of each layer in the current TPE model. As shown in Figure 6, each Transformer layer attends to different ranges, but the attention ranges of the same layer are similar across different images and target keypoints. In the earlier Transformer layers, attention is mainly distributed over the global region, where absolute positional information is more important. In the subsequent layers, attention gradually focuses on the target keypoint region, making relative positional information increasingly important. To enable each Transformer layer to learn more effective positional information, we further propose a layer-adaptive approach. This approach adjusts the contributions of absolute and relative positional information according to the attention distribution of each layer. Specifically, we assign learnable weight coefficients to both the absolute position embedding and the rotary position embedding of each layer:

$$\tilde{P}^m_{APE} = \lambda^m_{APE} \cdot P^m_{APE}, \tag{9}$$

$$\tilde{P}^m_{RoPE} = \lambda^m_{RoPE} \cdot e^{i\theta^m_{patch}}, \tag{10}$$

where $P^m_{APE} \in \mathbb{R}^{N_{patch} \times d}$ and $e^{i\theta^m_{patch}} \in \mathbb{R}^{N_{patch} \times d/2}$ are the absolute and rotary position embeddings of the patch tokens in the m-th Transformer layer. Their learnable layer-adaptive weights are λ^m_{APE} and λ^m_{RoPE}, respectively. $\tilde{P}^m_{APE} \in \mathbb{R}^{N_{patch} \times d}$ and $\tilde{P}^m_{RoPE} \in \mathbb{R}^{N_{patch} \times d/2}$ are the corresponding adjusted position embeddings.

IV. Analysis

In this section, we conduct an in-depth analysis of our proposed HTPE through attention maps. These attention maps represent pairwise similarities between tokens, reflecting their underlying relationships. By analyzing these maps, the interpretability of HTPE is enhanced. We visualize the attention by using color intensity to indicate the similarity between tokens. Since the final Transformer layer is closest to the output, it can directly reflect the model's final attention distribution, and therefore, we select the attention maps of this layer for visualization. Figure 7 presents both the keypoint token attention maps and the full token attention maps.

A. Keypoint Token Attention Map

As illustrated in Figure 7, we visualize the keypoint token attention interactions across different images, comparing the results before and after applying HTPE.

First, the attention interactions with HTPE show clear boundaries across different token pairs, whereas the baseline method produces more blurred separations (*e.g.,* the top-left region). In HPE, the relationship between a keypoint and different keypoints should exhibit differences. Therefore, HTPE demonstrates more discriminative attention patterns, better captures key regions, and more accurately models inter-keypoint relationships.

Second, the diagonal elements of the attention map represent the similarity of each keypoint with itself. Theoretically, these should stand out. With HTPE, attention along the diagonal is clearly emphasized for each token, while the baseline fails to highlight this effectively (*e.g.,* the bottom-right corner), indicating the attention distribution of HTPE is more reasonable.

Finally, in human structure, head keypoints are closely related to one another but only weakly connected to the body keypoints, while limb movement is relatively independent from the head. Compared with the baseline, HTPE yields clearer attention contours between head and body keypoints (*e.g.,* the bottom-left rectangle and the top-right rectangle), better reflecting this structural characteristic.

B. Full Token Attention Map

Similarly, we visualize the full token attention maps for different images in Figure 7. In the attention maps produced by HTPE, there are three clear diagonal stripes. This aligns with the inherent characteristic of human images, where each patch and keypoint tends to have higher similarity with itself and its neighboring regions. In contrast, the baseline method lacks these distinct stripes. Instead, it presents multiple blurry

diagonal stripes, suggesting that it fails to clearly distinguish the most important regions. Furthermore, the baseline shows artifacts in certain areas (*e.g.,* the top-right corner), while HTPE does not. These artifacts often represent anomalous regions with redundant information [49], suggesting that the attention distribution of HTPE is more rational.

In addition, we provide analyses of KP-RoPE, the learnable offset and scale factors, computational cost, the complex-valued formulation of q and k, and the transferability to other structured tasks in the supplementary material.

V. Experiments

In this section, we first evaluate the proposed HTPE on the COCO [18], CrowdPose [19], and OCHuman [20] datasets. Next, we conduct ablation studies on the components of our method to help better understand it. We then construct a challenging pose subset from MPII [50] with heavily rotated and non-upright samples to examine the effectiveness of HTPE under non-upright poses. Additionally, we analyze the failure cases of the proposed HTPE under challenging scenarios. Finally, we provide generalization evaluations of HTPE across different models and tasks, as well as visualization results under various scenarios in the supplementary material.

A. Implementation Details

1) Datasets: The COCO dataset consists of over 200k images and approximately 250k annotated human instances. Each instance is labeled with K = 17 keypoints that represent the human pose. Our models are trained on the COCO train2017 split, which contains 57k images. We individually evaluated the models on both val2017, which contains 5K images, and test-dev2017, which includes 20K images.

Compared to the COCO keypoint dataset, the CrowdPose dataset poses a greater challenge due to its large number of crowded and occluded scenes. It contains approximately 20k images and 80k annotated human instances, each labeled with 14 keypoints. The dataset is split into roughly 10k training, 2k validation, and 8k testing images.

The OCHuman dataset represents one of the most challenging benchmarks for crowded multi-person pose estimation, with an average MaxIoU (intersection over union of bounding boxes) of 0.67 per person. It comprises 4,731 images containing a total of 8,110 human instances. To ensure a fair comparison, we follow the evaluation protocol described in [20], training our models on the COCO training set and evaluating on the OCHuman test set.

MPII is a classic 2D human pose dataset collected from a wide range of real-world human activities, featuring diverse poses and complex articulations. It consists of roughly 25,000 images, containing about 40,000 human subjects, each labeled with 16 body keypoints.

We use the Rendered Hand Pose Dataset (RHD) [51] for 2D hand pose estimation to evaluate the cross-task adaptability of HTPE. The dataset contains 20 different characters performing 39 actions, split into a training set (16 characters, 31 actions) and a validation set (4 characters, 8 actions). It provides 41,258 training images and 2,728 validation images at 320×320 resolution. Each image is fully annotated with a 21-keypoint skeleton for each hand.

2) Evaluation Metric: For the COCO dataset, we adopt the standard Average Precision (AP) as the evaluation metric, which is calculated using Object Keypoint Similarity (OKS):

$$OKS = \frac{\sum_i exp(-d_i^2/2s^2 j_i^2)\sigma(v_i > 0)}{\sum_i \sigma(v_i > 0)} \tag{11}$$

where d_i is the Euclidean distance between the i-th predicted keypoint and its corresponding ground-truth location, j_i reprsents a per-keypoint constant, v_i denotes the visibility flag, σ denotes the indicator function, and s indicates the object scale. We follow the standard COCO evaluation protocol [18] and report standard average precision and recall scores: AP^{50} (average precision at OKS = 0.50), AP^{75} (average precision at OKS = 0.75), AP (the mean of average precision scores at 10 positions, OKS = 0.50, 0.55,..., 0.90, 0.95), AP^M for medium objects, AP^L for large objects, and AR (average recall) at OKS = 0.50, 0.55,..., 0.90, 0.95.

The evaluation on CrowdPose and OCHuman also adopts the standard COCO protocol, utilizing Average Precision (AP) based on Object Keypoint Similarity (OKS) as the evaluation metric. For the RHD dataset, we evaluate 2D keypoint detection using the Percentage of Correct Keypoints (PCK), considering a prediction correct if its distance to the ground truth is within the threshold of the output size. For the MPII dataset, we adopt the head-normalized probability of correct keypoint (PCKh) metric for evaluation. PCKh@0.5 and PCKh@0.1 indicate that a predicted keypoint is considered correct if it falls within 50% and 10% of the head size from the ground-truth keypoint, respectively, where the head size is defined as 60% of the diagonal length of the ground-truth head bounding box.

3) Settings and Training: We follow the top-down human pose estimation paradigm. The input images are resized to a resolution of 256×192. We adopt a commonly used human detector provided by SimpleBaselines [23], which achieves 56.4% AP on the COCO validation set and 60.9% AP on the COCO test-dev set.

a) Model setting: We apply HTPE to four models: SDPose-S-V1, SDPose-S-V2, SDPose-B, and SDPose-T.

Specifically, we replace the original absolute position embedding with the layer-adaptive hybrid patch position embedding and add the structure-aware keypoint position embedding. All other model configurations remain consistent with those in the SDPose [17] paper. SDPose-B uses HRNet-W32-stage3 as the backbone,while SDPose-S-V1, SDPose-S-V2, and SDPose-T use Stemnet as the backbone. SDPose-T contains only 6 Transformer layers, while SDPose-S-V1 and SDPose-S-V2 each contain 12 Transformer layers. The patch sizes are 4×3 for SDPose-S-V1 and 2×2 for SDPose-S-V2. We also apply HTPE to regression-based methods. Specifically, we integrate HTPE into SDPose-RLE, a variant of SDPose in which the original loss function is replaced by the RLE loss. In addition, to demonstrate the generality of HTPE, we also incorporate it into the TokenPose [14], ViTPose [8], and PPT [15] models.

b) Training detail: For training small models on the COCO dataset, we use 2 NVIDIA 2080ti GPUs, processing 32 samples per GPU. The optimizer used is Adam, with an initial learning rate of 1e-3, which is reduced to one-tenth of its

TABLE I
COMPARISON WITH HEATMAP-BASED METHODS ON THECOCO VALIDATION SET. THE INPUT IMAGE SIZE IS 256 × 192

Method	Backbone	Params(M)↓	GFLOPs↓	Speed(fps)↑	Memory(M)↓	AP↑	AR↑
SimpleBaseline [23]	ResNet-50	34.0	5.5	64.7	158.8	71.8	77.3
SimpleBaseline [23]	ResNet-101	53.0	9.1	36.6	231.7	72.6	78.1
SimpleBaseline [23]	ResNet-152	68.6	12.8	25.3	291.3	73.5	79.0
TokenPose-V1 [14]	Stemnet	6.6	2.4	51.6	63.8	69.5	74.9
TokenPose-V2 [14]	Stemnet	6.2	4.7	50.6	280.6	71.8	77.0
TokenPose [14]	HRNet-W32-Stage3	13.2	5.2	17.9	83.7	73.2	78.7
ViTPose-S [8]	ViT-S	24.3	5.3	59.5	100.7	73.8	79.2
ViTPose++-S [52]	ViT-S	42.0	8.7	42.4	172.9	75.8	82.6
MambaPose-S-V1 [53]	SS2D	-	2.8	-	-	72.8	78.2
MambaPose-S-V2 [53]	SS2D	-	4.0	-	-	74.2	79.6
CF-PGNN [54]	HRNet-W32	31.0	8.1	-	-	76.2	81.1
TransPose [39]	ResNet-Small	5.2	8.0	230.3	82.0	71.7	77.1
TransPose [39]	HRNet-Small-W48	17.5	21.8	32.0	252.5	75.8	80.8
PPT-V1 [15]	Stemnet	6.6	2.0	49.7	49.5	69.8	75.1
PPT [15]	HRNet-W32-Stage3	13.2	4.7	18.3	69.1	73.4	78.8
SDPose-T [17] (*baseline*)	Stemnet	4.4	1.8	77.9	42.5	69.7	75.2
+HTPE	Stemnet	4.4	1.8	63.6	44.5	71.0$_{+1.3}$	76.4
SDPose-V1 [17] (*baseline*)	Stemnet	6.6	2.4	51.3	65.8	72.3	77.7
+HTPE	Stemnet	6.6	2.4	41.9	68.8	74.1$_{+1.8}$	79.4
SDPose-V2 [17] (*baseline*)	Stemnet	6.2	4.7	50.5	286.3	73.5	78.7
+HTPE	Stemnet	6.2	4.7	41.5	295.0	**75.1**$_{+1.6}$	**80.4**
SDPose [17] (*baseline*)	HRNet-W32-Stage3	13.2	5.2	18.6	85.7	73.7	79.1
+HTPE	HRNet-W32-Stage3	13.2	5.2	17.0	89.4	74.7$_{+1.0}$	80.0

TABLE II
COMPARISON WITH REGRESSION-BASED METHODS ON THE COCO VALIDATION SET. MODELS ARE RETRAINED AND EVALUATED ON MMPOSE

Method	Backbone	Input Size	AP↑	Params(M)↓	GFLOPs↓	Speed(fps)↑	Memory(M)
PRTR [12]	ResNet-50	384×288	68.2	41.5	11.0	60.5	350.5
PRTR [12]	ResNet-50	512×384	71.0	41.5	18.8	63.4	373.3
Poseur [11]	MobileNet	256×192	71.9	11.4	0.5	17.1	111.2
Poseur [11]	ResNet-50	256×192	75.4	33.3	4.6	16.3	240.0
RLE [10]	ResNet-50	256×192	70.5	23.6	4.0	86.7	117.1
DistilPose-V1 [16]	Stemnet	256×192	71.6	5.4	2.4	61.1	34.6
SDPose-RLE [17]	Stemnet	256×192	72.1	5.7	2.4	52.9	62.2
+HTPE	Stement	256×192	**73.8**$_{+1.7}$	5.7	2.4	43.7	65.2

original value at the 200th and 260th epochs, respectively. All tokenized pose estimation methods are trained for 300 epochs.

For larger models, we use 2 NVIDIA A40 GPUs. The batch size is set to 384 for ViTPose-S and 64 for ViTPose-B. We adopt a 256 × 192 input resolution and use the AdamW optimizer with a learning rate of 5e-4. The models are trained for 210 epochs, with the learning rate reduced by a factor of 0.1 at the 170th and 200th epochs, respectively.

Due to the small-scale experimental validation, the number of training epochs on RHD is set to 120, while all other training settings for CrowdPose, MPII, and RHD follow those used for the COCO dataset. The speed and memory usage of all methods were measured with a batch size of 1: on a single 2080Ti GPU for lightweight models and on a single A40 GPU for larger models.

B. Main Results

1) Evaluation on COCO Dataset: We present a comparison of HTPE applied to SDPose [17] with the SOTA models on the COCO validation set and the COCO test dataset. The experimental results are shown in Table I, Table II and Table III, respectively. Whether compared to heatmap-based methods or regression-based methods, our approach achieves SOTA performance among lightweight models. SDPose with HTPE achieves better performance compared to the baseline methods with almost no increase in model parameters and GFLOPs. Specifically, compared to SDPose [17], SDPose with HTPE achieves significant performance improvements in AP on Stemnet-T, Stemnet-V1, Stemnet-V2, and HRNet-W32-stage3 [56] models, with gains of 1.3%, 1.8%, 1.6%, and 1.0%, respectively. In particular, SDPose-T with HTPE achieves 71.0 AP with only 4.4M parameters and 1.8 GFLOPs, delivering results comparable to other small models while requiring significantly fewer parameters and computations. Furthermore, SDPose-RLE with HTPE achieves 73.8 AP with 5.7M and 2.4 GFLOPs, making it the most efficient regression-based pose estimation model. Similarly, our approach achieves SOTA performance on the COCO test set, showing notable improvements over the baseline. As shown in Table III, our method consistently outperforms baseline models of different sizes, achieving improvements of 1.3%, 2.4%, and 1.7%, respectively. When applied to regression models, our method also achieves a 1.5% improvement. Although there is a slight decrease in speed and a minor increase in memory footprint, our method improves model performance with minimal additional parameters. In summary, these results provide strong evidence for the effectiveness of HTPE.

2) Evaluation on Crowdpose Dataset: To demonstrate the generalization and occlusion robustness of our method, we conducted experiments on the CrowdPose dataset. As shown in Table IV, SDPose with HTPE consistently outperforms the baseline across different models and achieves SOTA performance. Specifically, SDPose-S-V1 with HTPE surpasses the baseline by 0.9 AP, while SDPose-S-V2 with HTPE improves it by 3.3 AP. ***The performance improvement under occlusion***

TABLE III
COMPARISON ON MSCOCO TEST-DEV DATASET. MODELS ARE RETRAINED AND EVALUATED ON MMPOSE

Method	Type	Input Size	Params(M)↓	GFLOPs↓	Speed(fps)↑	Memory(M)↓	AP↑	AP_{50}↑	AP_{75}↑	AP_M↑	AP_L↑
TokenPose-S-V1 [14]	heatmap -based	256×192	6.6	2.4	51.6	63.8	68.6	89.9	76.1	65.1	74.5
TokenPose-S-V2 [14]		256×192	6.6	4.7	50.6	280.6	71.1	90.4	78.7	67.7	77.1
PPT-S-V1 [15]		256×192	6.6	2.0	49.7	49.5	69.2	90.1	76.8	65.8	75.2
SDPose-T [17] (*baseline*)		256×192	4.4	1.8	77.9	42.5	69.2	90.2	76.8	65.7	75.2
+HTPE		256×192	4.4	1.8	63.6	44.5	70.5$_{+1.3}$	90.9	78.1	67.1	76.5
SDPose-S-V1 [17] (*baseline*)		256×192	6.6	2.4	51.3	65.8	71.7	91.1	79.5	68.3	77.5
+HTPE		256×192	6.6	2.4	41.9	68.8	74.1$_{+2.4}$	92.0	81.9	70.8	80.0
SDPose-S-V2 [17] (*baseline*)		256×192	6.2	4.7	50.5	286.3	72.7	91.2	80.3	69.3	78.5
+HTPE		256×192	6.2	4.7	41.5	295.0	**74.4**$_{+1.7}$	**92.0**	**82.1**	**71.1**	**80.3**
RLE-Res50 [10]		256×192	23.6	4.0	86.7	117.1	69.8	90.1	77.5	67.2	74.3
DistilPose-S-V1 [16]		256×192	5.4	2.4	61.1	34.6	71.0	91.0	78.9	67.5	76.8
SDPose-RLE [17] (*baseline*)		256×192	5.7	2.4	52.9	62.2	71.4	90.5	78.9	68.0	76.9
+HTPE		256×192	5.7	2.4	43.7	65.2	**72.9**$_{+1.5}$	**90.9**	**80.3**	**69.6**	**78.5**

TABLE IV
COMPARISON WITH SOTA METHODS ON CROWDPOSE TEST SET. THE INPUT IMAGE SIZE IS 256 × 192

Method	AP↑	Params(M)↓	GFLOPs↓	Speed(fps)↑	Memory(M)↓
SimpleBaseline-Res50 [23]	63.7	34.0	8.9	64.7	158.8
KAPAO-S [55]	63.8	12.6	-	43.6	109.2
PPT-S [15]	55.6	6.6	2.0	49.7	49.5
RLE-Res50 [10]	57.0	23.6	4.0	6.7	117.1
SDPose-S-V1 [17]	64.5	6.6	2.4	51.3	65.8
+HTPE	65.4$_{+0.9}$	6.6	2.4	41.9	68.8
SDPose-S-V2 [17]	63.8	6.2	4.7	50.5	286.3
+HTPE	**67.1**$_{+3.3}$	6.2	4.7	41.5	295.0

TABLE V
RESULTS ON OCHUMAN TEST SET. THE INPUT IMAGE SIZE IS 256 × 192. ALL MODELS ADOPT STEMNET AS THE BACKBONE, AND THEIR TRANSFORMER CONFIGURATIONS ARE CONSISTENT WITH SDPOSE-S-V1. MODELS ARE RETRAINED AND EVALUATED ON MMPOSE

Method	AP↑	Params(M)↓	GFLOPs↓	Speed(fps)↑	Memory(M)↓
TokenPose [14]	58.2	6.6	2.4	51.6	63.8
+HTPE	59.6$_{+1.4}$	6.6	2.4	37.9	66.6
PPT [15]	57.1	6.6	2.0	49.7	49.5
+HTPE	59.8$_{+2.7}$	6.6	2.0	35.6	50.0
SDPose [17]	59.4	6.6	2.4	51.3	65.8
+HTPE	59.5$_{+0.1}$	6.6	2.4	41.9	68.8

conditions demonstrates the effectiveness of our human structural modeling, as the human visual system regards body structure as an effective prior for addressing occlusion.

3) Evaluation on OCHuman Dataset: To thoroughly validate the effectiveness of our method under occlusion, we further conducted experiments on the OCHuman dataset. The OCHuman dataset is a more challenging dataset involving severe occlusions. As shown in Table V, HTPE improves the performance of all evaluated models and achieves SOTA results. In particular, TokenPose equipped with HTPE outperforms its baseline by 1.4 AP, whereas PPT improves by of 2.7 AP. We observe that adding HTPE results in only a 0.1 AP improvement for SDPose, while the performance of SDPose, TokenPose+HTPE, and PPT+HTPE is comparable. This suggests that SDPose has already fully exploited the potential of the current lightweight Transformer architecture, reaching performance saturation. Consequently, further incorporating structural and positional information yields limited improvements. In summary, experimental results on the OCHuman dataset further demonstrate the robustness of our proposed HTPE to occlusions.

TABLE VI
ABLATION STUDIES FOR EACH MODULE. ALL ABLATION EXPERIMENTS ARE BASED ON SDPOSE-S-V1, IMPROV. = IMPROVEMENT

APE	LHPPE	SAKPE	AP↑	Improv.	Speed(fps)↑	Memory(M)↓
✓			72.3	-	51.3	65.8
		✓	73.0	+0.7	43.0	66.4
	✓		73.6	+1.3	41.3	69.4
	✓	✓	**74.1**	+1.8	41.9	68.8

C. Ablation Study

1) Major Modules: We conduct several ablation experiments to verify the effectiveness of major modules in HTPE. As shown in Table VI, all proposed modules benefit our model. Specially, SAKPE alone improves performance by 0.7%. In addition, when SAKPE is used alone on PPT-S, it achieves an improvement of 3.2% (see Figure colorblue12). It fully demonstrates the effectiveness and generality of SATPE. Furthermore, LHPPE brings an improvement of 1.3%. All modules together bring the best performance, which significantly improves the performance by 1.8%. Overall, each module plays a role and they work collaboratively to effectively capture and utilize the positional information of the tokens.

To clarify the individual contribution of each component, we perform ablation studies on every component within both SAKPE and LHPPE.

2) Keypoint Token Coordinate Encoding in SAKPE: We conduct an ablation study on the Keypoint Token Coordinate Encoding (KTCE) to specifically evaluate its standalone effectiveness. For a fair comparison, all original human keypoint coordinates are replaced with a constant value of 1.0, while

TABLE VII

ABLATION STUDIES FOR THE KEYPOINT TOKEN COORDINATE ENCODING (KTCE) OF SAKPE

Method	KTCE	AP↑	AR↑	Speed(fps)↑	Memory(M)↓
SAKPE	×	73.4	78.8	41.5	68.8
SAKPE	✓	**74.1**	**79.4**	41.9	68.8

TABLE VIII

ABLATION STUDIES FOR LEARNABLE FACTORS ξ AND β OF SAKPE

Method	ξ	β	AP↑	Speed(fps)↑	Memory(M)↓
SAKPE	×	×	72.9	41.8	68.8
SAKPE	✓	×	73.8	41.8	68.8
SAKPE	✓	✓	**74.1**	41.9	68.8

TABLE IX

ABLATION STUDIES FOR THE K-ROPE OF SAKPE

Method	K-RoPE	AP↑	AR↑	Speed(fps)↑	Memory(M)↓
SAKPE	×	73.7	79.1	38.6	69.7
SAKPE	✓	**74.1**	**79.4**	41.9	68.8

TABLE X

ABLATION STUDIES FOR P-ROPE OF LHPPE

Method	P-RoPE	AP↑	AR↑	Speed(fps)↑	Memory(M)↓
LHPPE	×	73.1	78.5	40.9	66.6
LHPPE	✓	**74.1**	**79.4**	41.9	68.8

keeping all other settings unchanged. Since multiplying any value by 1.0 does not alter its magnitude, this modification effectively removes the human structure information, forming a proper baseline without KTCE. As shown in Table VII, incorporating KTCE leads to a 0.7% improvement. This demonstrates that KTCE is an indispensable component of HTPE, providing critical structural cues that significantly contribute to the overall performance.

3) Learnable Parameters ξ and β in SAKPE: We carry out ablation analyses on the learnable parameters ξ and β in SAKPE. Table VIII shows that the introduction of learnable factors ξ and β individually improves the model's performance. These results further quantitatively validate the effectiveness of the learnable scale and offset factors in adapting to various pose variations.

4) K-RoPE in SAKPE: To assess the individual contribution of K-RoPE, we perform an ablation study. Specifically, the Keypoint-based Rotary Position Embedding (K-RoPE) is substituted with the original direct addition mechanism, while all other configurations remain unchanged. In this baseline, the keypoint position encodings ($\tilde{x}_{kpt}$, $\tilde{y}_{kpt}$) are directly added to the keypoint tokens [17]. The performance of this baseline is then compared with the full K-RoPE implementation. As illustrated in Table IX, incorporating K-RoPE alone leads to a 0.4% improvement in AP, corroborating its efficacy in enhancing the model's performance.

TABLE XI

ABLATION STUDIES FOR THE LEARNABLE PARAMETERS ω AND λ OF LHPPE

Method	ω	λ	AP↑	AR↑	Speed(fps)↑	Memory(M)↓
LHPPE	✓	×	73.9	79.2	42.9	68.6
LHPPE	×	✓	73.0	78.4	40.9	69.0
LHPPE	✓	✓	**74.1**	**79.4**	41.9	68.8

TABLE XII

COMPARISON OF DIFFERENT POSITION EMBEDDINGS. **"APE (PATCH)" DENOTES THAT THE ABSOLUTE POSITION EMBEDDING IS APPLIED ONLY TO PATCH TOKENS, WHILE OTHER POSITION EMBEDDING METHODS ARE APPLIED TO ALL TOKENS**

Position embedding	AP↑	AR↑	Latency(ms)↓	Memory(M)↓	Params(M)↓	GFLOPs↓
APE (patch) [42]	72.3	77.7	19.5	65.8	6.6	2.4
APE [42]	72.0	77.4	20.4	65.8	6.6	2.4
RPE [43]	70.8	76.4	20.2	95.7	13.8	2.4
RoPE [46]	73.8	79.1	23.9	66.4	6.6	2.4
iRPE [44]	72.9	78.5	22.8	73.6	6.7	2.4
KP-RoPE	**74.1**	**79.4**	23.9	68.8	6.6	2.4

5) P-RoPE in LHPPE: To investigate the effect of P-RoPE in isolation, we perform an ablation study by omitting it from the model. As shown in Table X, removing P-RoPE causes a 1.1% decrease in performance. This highlights the significance of P-RoPE, which effectively encodes the positional relationships among patch tokens.

The computational overhead of P-RoPE and K-RoPE is very small, even smaller than that of the slicing operations used to extract keypoint or patch tokens. As a result, removing P-RoPE or K-RoPE leads to longer computation time than HTPE.

6) Learnable Parameters ω and λ in LHPPE: Table XI demonstrates that the learnable frequency ω in LHPPE significantly enhances the model's performance. Specifically, the learnable frequency ω leads to a 1.1% improvement in AP. The results suggest that the learnable frequency facilitates the integration of 2D information. Furthermore, introducing the layer-adaptive weights λ can also improve the model's performance. It indicates that adaptively learning absolute and relative positional information according to the attention distribution of different layers can further exploit the potential of token positional relationships.

7) Position Embedding: To validate the effectiveness of our proposed KP-RoPE, we conduct experiments comparing it with different position embeddings, including APE [42], RPE [43], RoPE [46], iRPE [44]. For APE, we encode both patch coordinates and keypoint token coordinates into sinusoidal positional embeddings in a unified manner. For RPE, we employed a learnable bias matrix to capture the relative positional information between tokens. For RoPE, we simply apply the rotary position embedding described in RoFormer [46] to both patch tokens and keypoint tokens, without any adaptive fusion between the X-axis and Y-axis. For iRPE, considering the full interaction between keypoint tokens and patch tokens, as well as the fusion of the X-axis and Y-axis information, we adopt the contextual mode based on the product method. As shown in Table XII, KP-RoPE reaches 74.1 AP, while APE, RPE, RoPE, and iRPE reach 72.0 AP, 70.8 AP, 73.8 AP,

TABLE XIII
RESULTS OF OUR METHOD ON THE NON-UPRIGHT POSE DATASET. THE INPUT IMAGE SIZE IS 256 × 192. ALL MODELS ADOPT STEMNET AS THE BACKBONE

Method	PCKh@0.5↑	PCKh@0.1↑	Params(M)↓	GFLOPs↓	Speed(fps)↑	Memory(M)↓
SDPose-V1 [17]	64.1	5.3	6.6	2.4	51.3	65.8
+HTPE	64.2	$6.5_{+1.2}$	6.6	2.4	41.9	68.8

72.9 AP, respectively. Our proposed KP-RoPE outperforms other position embeddings. This can be attributed to the fact that KP-RoPE not only effectively learns relative positional information but also better facilitates the coupling between patch position embedding and keypoint position embedding. From the perspective of inference cost, our method has the smallest number of parameters and the lowest GFLOPs, while incurring slightly higher memory consumption and inference latency. Overall, our method prioritizes computational efficiency and model compactness, achieving state-of-the-art performance at the cost of only a minor increase in memory consumption and inference latency.

D. Robustness to Non-Upright Poses.

We evaluate the performance of HTPE on a non-upright pose dataset, which is constructed by selecting non-upright samples from the MPII [50] validation set. The MPII dataset covers diverse human poses and a wide range of human activities (over 800 categories), with all images collected from complex real-world scenarios (sourced from YouTube videos). Therefore, the constructed non-upright pose dataset provides sufficient data coverage and is more aligned with real-world scenarios. The constructed non-upright pose dataset contains 65 images and almost covers all non-upright pose samples in the MPII validation set. It encompasses a diverse set of poses, including handstands, gymnastics, yoga, lying, and swimming. As shown in Table XIII, when the threshold is set to 0.1, HTPE achieves an improvement of approximately +1.2. Notably, when the threshold is set to 0.5, the improvement brought by HTPE is only +0.1, indicating relatively limited gains. Therefore, under large pose rotations, HTPE mainly improves the localization accuracy of easy keypoints, while the improvement on challenging keypoints remains limited, indicating that there is still room for further enhancement in modeling structural priors under such scenarios. In conclusion, although upright poses dominate real-world scenarios and HTPE performs well in such cases, non-upright poses still pose certain challenges.

Additional visualizations are provided in the supplementary material. In future work, we will explore more flexible modeling strategies, such as rotation-aware coordinate systems, to further improve performance on non-upright poses.

E. Failure Analysis

Figure 8 presents the failure cases of our HTPE. Our approach mainly exhibits two types of failures. The first type arises in severely crowded scenes. In such cases, the large number of people and their significant overlap make it hard for the detector to accurately locate each human bounding box. Moreover, the occlusions introduced by crowding further hinder the pose estimation process. Therefore, dealing with severe crowding remains an open challenge. The second type of failure arises in heavily occluded scenes, where the human body is obscured not only by other people but also by various objects. In many cases, even human observers may find it difficult to accurately identify the pose. In future work, we plan to tackle these challenges by leveraging the extensive world knowledge of multimodal large models and employing more advanced human modeling techniques.

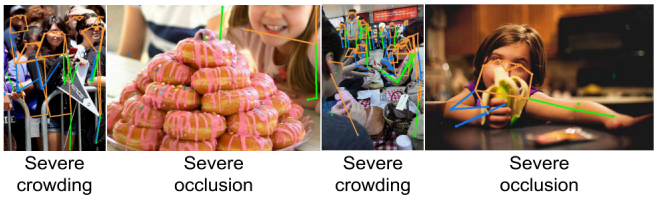

Fig. 8. Failure cases of our method. Example images are taken from the CrowdPose dataset.

VI. CONCLUSION

In this paper, we propose a human-structure-aware token position embedding for tokenized pose estimation, HTPE. It injects human body structure priors into the keypoint token position embedding, fully leveraging the inherent relative positional relationships between keypoints. Additionally, it adaptively learns relative and absolute positional information based on the attention distribution of different Transformer layers, further capturing the spatial relationships among patches. Extensive experiments demonstrate the effectiveness of the proposed HTPE method. In short, HTPE achieves SOTA performance among lightweight HPE models with almost a slight increase in computational cost. We believe that encoding structure characteristics into position embeddings may inspire other vision tasks, such as image segmentation, object detection, and human action recognition, *etc.*

REFERENCES

[1] Z. Geng, C. Wang, Y. Wei, Z. Liu, H. Li, and H. Hu, "Human pose as compositional tokens," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 660–671.
[2] T. Wang et al., "DecenterNet: Bottom-up human pose estimation via decentralized pose representation," in *Proc. 31st ACM Int. Conf. Multimedia (ACM MM)*, Oct. 2023, pp. 1798–1808.
[3] T. Wang et al., "SynSP: Synergy of smoothness and precision in pose sequences refinement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 1824–1833.
[4] W. Li et al., "HYRE: Hybrid regressor for 3D human pose and shape estimation," *IEEE Trans. Image Process.*, vol. 34, pp. 235–246, 2025.
[5] F. Tian and S. Kim, "LEAPSE: Learning environment affordances for 3D human pose and shape estimation," *IEEE Trans. Image Process.*, vol. 33, pp. 3285–3300, 2024.
[6] M. T. Hassan and A. Ben Hamza, "Regular splitting graph network for 3D human pose estimation," *IEEE Trans. Image Process.*, vol. 32, pp. 4212–4222, 2023.
[7] L. Jin et al., "Single-stage is enough: Multi-person absolute 3D pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13086–13095.
[8] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose: Simple vision transformer baselines for human pose estimation," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 38571–38584.

[9] J. Wang et al., "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

[10] J. Li et al., "Human pose regression with residual log-likelihood estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 11025–11034.

[11] W. Mao et al., "Poseur: Direct human pose regression with transformers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2022, pp. 72–88.

[12] K. Li, S. Wang, X. Zhang, Y. Xu, W. Xu, and Z. Tu, "Pose recognition with cascade transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1944–1953.

[13] L. Zhao, N. Wang, C. Gong, J. Yang, and X. Gao, "Estimating human pose efficiently by parallel pyramid networks," *IEEE Trans. Image Process.*, vol. 30, pp. 6785–6800, 2021.

[14] Y. Li et al., "Tokenpose: Learning keypoint tokens for human pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 11313–11322.

[15] H. Ma et al., "PPT: Token-pruned pose transformer for monocular and multi-view human pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2022, pp. 424–442.

[16] S. Ye et al., "DistilPose: Tokenized pose regression with heatmap distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 2163–2172.

[17] S. Chen et al., "SDPose: Tokenized pose estimation via circulation-guide self-distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 1082–1090.

[18] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 740–755.

[19] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, and C. Lu, "CrowdPose: Efficient crowded scenes pose estimation and a new benchmark," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10863–10872.

[20] S.-H. Zhang et al., "Pose2Seg: Detection free human instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 889–898.

[21] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab, "Robust optimization for deep regression," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2830–2838.

[22] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1653–1660.

[23] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 466–481.

[24] F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu, "Distribution-aware coordinate representation for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7091–7100.

[25] J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, Aug. 2014, pp. 1–11.

[26] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7103–7112.

[27] Y. Li et al., "Simcc: A simple coordinate classification perspective for human pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2022, pp. 89–106.

[28] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.

[29] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *Proc. CVPR*, Jun. 2011, pp. 1385–1392.

[30] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Poselet conditioned pictorial structures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 588–595.

[31] X. Chu, W. Ouyang, H. Li, and X. Wang, "Structured feature learning for pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4715–4723.

[32] Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang, "Adversarial posenet: A structure-aware convolutional network for human pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1212–1221.

[33] D. S. Raychaudhuri, C.-K. Ta, A. Dutta, R. Lal, and A. K. Roy-Chowdhury, "Prior-guided source-free domain adaptation for human pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 14950–14960.

[34] N. Yoo and O. Russakovsky, "Efficient, self-supervised human pose estimation with inductive prior tuning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2023, pp. 3263–3272.

[35] Z. Wang, S. Han, and M. Zhang, "Pose prior learner: Unsupervised categorical prior learning for pose estimation," 2024, *arXiv:2212.05262*.

[36] J. Peng, Y. Zhou, and P. Mok, "KTPFormer: Kinematics and trajectory prior knowledge-enhanced transformer for 3D human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 1123–1132.

[37] L. Han, K. Chen, L. Zhao, Y. Jiang, P. Wang, and N. Zheng, "Cross-domain animal pose estimation with skeleton anomaly-aware learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 9, pp. 9148–9160, Sep. 2025.

[38] W. Mao, Y. Ge, C. Shen, Z. Tian, X. Wang, and Z. Wang, "Tfpose: Direct human pose estimation with transformers," 2021, *arXiv:2103.15320*.

[39] S. Yang, Z. Quan, M. Nie, and W. Yang, "Transpose: Keypoint localization via transformer," in *Proc. EEE Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 11802–11812.

[40] J. Soucie et al., "Range of motion measurements: Reference values and a database for comparison studies," *Haemophilia*, vol. 17, no. 3, pp. 500–507, Nov. 2011.

[41] V. F. Ferrario, C. Sforza, G. Serrao, G. Grassi, and E. Mossi, "Active range of motion of the head and cervical spine: A three-dimensional investigation in healthy young adults," *J. Orthopaedic Res.*, vol. 20, no. 1, pp. 122–129, Jan. 2002.

[42] A. Vaswani, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–12.

[43] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," 2018, *arXiv:1803.02155*.

[44] K. Wu, H. Peng, M. Chen, J. Fu, and H. Chao, "Rethinking and improving relative position encoding for vision transformer," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10033–10041.

[45] R. Yu et al., "Position embedding needs an independent layer normalization," 2022, *arXiv:2212.05262*.

[46] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "RoFormer: Enhanced transformer with rotary position embedding," *Neurocomputing*, vol. 568, Feb. 2024, Art. no. 127063.

[47] B. Heo, S. Park, D. Han, and S. Yun, "Rotary position embedding for vision transformer," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2024, pp. 289–305.

[48] Z. Lu et al., "FiT: Flexible vision transformer for diffusion model," 2024, *arXiv:2402.12376*.

[49] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski, "Vision transformers need registers," 2023, *arXiv:2309.16588*.

[50] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3686–3693.

[51] C. Zimmermann and T. Brox, "Learning to estimate 3D hand pose from single RGB images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4913–4921.

[52] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose++: Vision transformer for generic body pose estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 2, pp. 1212–1230, Feb. 2024.

[53] Y. Xu, M. Jiang, Y. Gao, J. Mu, D. Wang, and L. Zhao, "MambaPose: Efficient 2D human pose estimation with pose-prior guided state space model," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jun. 2025, pp. 1–6.

[54] Z. Ji, W. Zhang, S. Qiao, K. Feng, and Y. Qian, "A coarse-to-fine human pose estimation method based on two-stage distillation and progressive graph neural network," 2025, *arXiv:2508.11212*.

[55] W. McNally, K. Vats, A. Wong, and J. McPhee, "Rethinking keypoint representations: Modeling keypoints and poses as objects for multi-person human pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2022, pp. 37–54.

[56] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5693–5703.