Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.059006

ARTICLE

Token Masked Pose Transformers Are Efficient Learners

Xinyi Song[1], **Haixiang Zhang**[1,*] **and Shaohua Li**[2]

[1]School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou, 310018, China
[2]College of Artificial Intelligence, Nankai University, Tianjin, 300350, China

*Corresponding Author: Haixiang Zhang. Email: zhhx@zstu.edu.cn

Received: 25 September 2024; Accepted: 10 January 2025; Published: 16 April 2025

ABSTRACT: In recent years, Transformer has achieved remarkable results in the field of computer vision, with its built-in attention layers effectively modeling global dependencies in images by transforming image features into token forms. However, Transformers often face high computational costs when processing large-scale image data, which limits their feasibility in real-time applications. To address this issue, we propose Token Masked Pose Transformers (TMPose), constructing an efficient Transformer network for pose estimation. This network applies semantic-level masking to tokens and employs three different masking strategies to optimize model performance, aiming to reduce computational complexity. Experimental results show that TMPose reduces computational complexity by 61.1% on the COCO validation dataset, with negligible loss in accuracy. Additionally, our performance on the MPII dataset is also competitive. This research not only enhances the accuracy of pose estimation but also significantly reduces the demand for computational resources, providing new directions for further studies in this field. Code is available at: https://github.com/lshua98/tmpose (accessed on 9 January 2025).

KEYWORDS: Pattern recognition; image processing; neural network; pose transformer

1 Introduction

The main task of Two-Dimensional (2D) human pose estimation is to accurately locate the coordinates of key points on the human body in a given 2D image. Human pose estimation has become a critical topic within computer vision research due to its significant potential across various real-world applications, such as action recognition [1–3], human-computer interaction [4–6], and sports analytics. The problem involves accurately locating human joints and body parts in images, a task made challenging by variations in body shape, appearance, and complex poses. Recent advancements in deep learning have greatly improved the performance of pose estimation models, enhancing both speed and accuracy to levels suitable for deployment. As deep neural networks evolve, they continue to drive new possibilities in understanding and predicting human motion, making it an increasingly impactful field of study.

In the past decade, deep Convolutional Neural Networks (CNNs) have dominated the field of human pose estimation [7–11]. However, researchers quickly discovered that unlike fully connected networks, CNNs are sparse connectivity networks with local convolutional properties. As a result, they fail to effectively capture the global dependency information of images. For complex detection tasks such as human pose estimation, it is crucial to fully capture the information embedded in the images.

Transformer is a sequence model, has achieved significant success in Natural Language Processing (NLP) [12–16]. The main mechanism of the Transformer is the multi-head self-attention layer. Compared

to convolution, it can effectively model the global dependency relationships of image features. Many studies have indicated [17–19] that the Transformer has a higher upper limit and can achieve better results than CNNs. As a result, the Transformer has gradually been introduced to the computer vision and has demonstrated great potential [19–22]. In human pose estimation, Learning Keypoint Tokens for Human Pose Estimation (TokenPose) [23] has achieved state-of-the-art performance among networks of the same type. The authors extract visual features using Convolutional Neural Networks (CNN) and then rearrange these visual features into One-Dimensional (1D) tokens, which are fed into Transformers for feature learning. This paradigm has gained consensus, but its drawbacks are also apparent. The quantity and length of the token sequence greatly affect the complexity of the model. These drawbacks become even more evident in scenarios with high-resolution input images or multiple views. To reduce the complexity of the model, relevant works often adopt the strategy of pruning the backbone network to balance the computational cost of the Transformer. For instance, in TokenPose [23], the authors only utilize the first three stages of High-Resolution Net (HRNet) while discarding the computationally expensive fourth stage. Although this approach enables the entire model to achieve the highest accuracy while maintaining desirable computational complexity, the author believes that this reduction in computational complexity is achieved through pruning the CNN network, while the fundamental nature of the Transformer remains unchanged. Therefore, the author began to contemplate whether it is possible to truly reduce the overall complexity of a model by lowering the complexity of the Transformer?

Multitask Auto-Encoder (MAE) [24] proposed a self-supervised method that predicts the original image by randomly masking image patches. In addition, Bidirectional Encoder representation from Image Transformers (BEiT) [25] also introduced a self-supervised method that masks tokens to pretrain the Vision Transformer. These two works have greatly inspired our study. Through analysis, we found that the computational complexity of Transformers is determined by tokens. Therefore, we propose a token-masked Transformer for pose estimation, as illustrated in Fig. 1. The rationale behind this approach can be explained from the following perspectives:

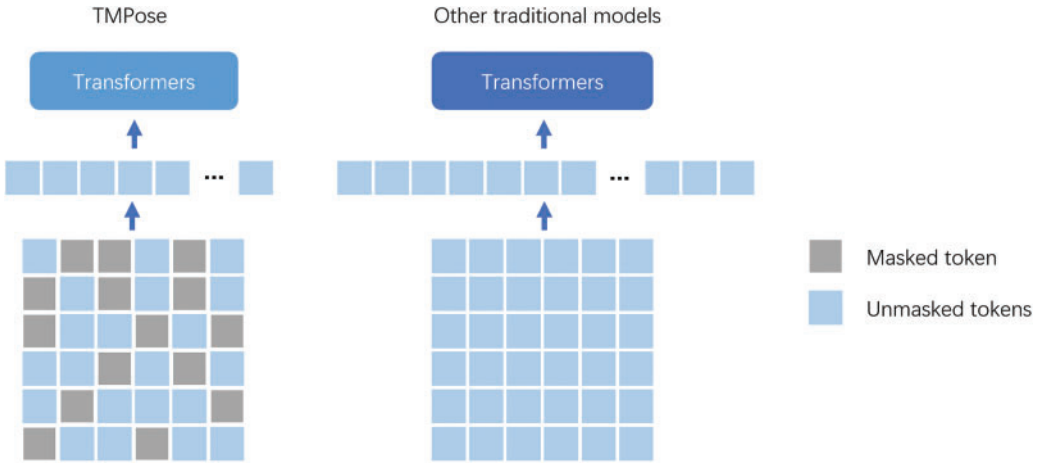

Figure 1: The core idea of TMPose is presented in left figure. Due to the adoption of only a subset of tokens, the computational complexity of the Transformer is significantly reduced. Right figure illustrates a series of Transformer-based human pose estimation networks such as TokenPose. Since all tokens are utilized, the computational complexity of the Transformer component is substantially increased

1. Text is a highly condensed artificial symbol, while images are natural signals that contain significant information redundancy. Additionally, images also contain a considerable amount of noise information. Therefore, we employ random sampling to input a portion of tokens into the network for training.

2. For images, a missing patch does not significantly affect the global semantics, and it is relatively easy to restore the missing region using the surrounding pixel information.

3. From the perspective of Bionics, for an image, the human eye can infer its related information without observing all of it. Therefore, for Transformer, it does not need all tokens and also can achieve a good result in theory.

Specifically, this paper takes TokenPose [23] as the baseline. Firstly, feature extraction by using CNNs. Then, the feature maps are tokenized, and we employ three different methods for handling the generated tokens: random sampling, uniform sampling, and block sampling. The tokenized data is subsequently fed into the Transformer network for feature learning. Finally, keypoints are predicted by heatmap-based approach. In summary, the main contributions of this paper are as follows:

- We propose a novel approach for efficient pose estimation, named TMPose, by selectively masking a portion of tokens. The idea of employing masks in pose estimation is relatively uncommon, as the author's knowledge.
- We proposed three different mask types and multiple mask rates to reduce computational complexity, and conducted nine sets of cross experiments to verify the efficiency of TMPose
- We demonstrate TMPose's competitive edge by achieving a 61.1% reduction in GFLOPs^{T} on public benchmark datasets, while maintaining high accuracy. These results highlight TMPose's efficiency and effectiveness in terms of both speed and accuracy

2 Related Work

2.1 Efficient Vision Transformers

In recent years, significant progress has been made in vision Transformers. Representative works include image classification [19], object detection [22], and semantic segmentation [21,26]. Although these models demonstrate superior accuracy compared to CNN-based models, such as Vision Transformer (ViT) [19], they come at a high computational cost. As a result, researchers have started proposing various algorithms to improve the efficiency of these models. For instance, techniques like pruning [27,28], distillation [29,30], and quantization [31,32] have been introduced in model compression to enhance efficiency.

In addition to these conventional methods, some researchers have proposed token pruning as an alternative approach. In self-supervised learning, several efficient token pruning methods have achieved promising results. MAE [24] introduced a self-supervised model with an encoder-decoder structure. The model is trained by randomly masking patches in the image as labels for supervision. Fast Language-Image Pre-training (Flip) [33] masked image patches and only encoded the visible patches, and then trained the model through contrastive learning with text samples. In BEiT [25], the authors proposed a masked image modeling approach for pretraining ViT models. This method efficiently trains the ViT model by pixel-level masking of image patches and semantic-level masking of visual tokens.

After the completion of pre-training, the idea of token pruning has been reflected in many works when applied to downstream tasks [34–36]. Specifically, Token-to-Tokens [34] concatenates adjacent tokens into a single token to reduce the total number of tokens. Efficient Vision Transformers with Dynamic Token Sparsification (DynamicViT) [35] employs a learnable token selector to prune tokens, while in token-Pruned Pose Transformer (PPT) [36], human token identification (HTI) is introduced, and token pruning is achieved through attention-based operations. Although these methods have achieved impressive results, they add

additional learnable parameters into the Transformer architecture to reduce the number of tokens and thus decrease model complexity. However, in TMPose, we do not introduce any extra learnable parameters while still achieving excellent performance.

2.2 Pose Transformers

From Section 2.1, it is evident that Transformer has witnessed rapid development in the field of computer vision. Similarly, in the task of human pose estimation, numerous remarkable works have emerged [37–41]. Transformer for human pose (TransPose) [37] was the first to introduce Transformer into human pose estimation. In this model, the features are initially extracted using a backbone network, and then the feature maps are transformed into token form and fed into the Transformer and keypoint predictions are made using heatmap regression. TransPose achieved state-of-the-art (SOTA) performance at that time through this simple framework. Subsequently, Transformer-based Pose estimation (TFPose) [38] built upon TransPose by eliminating the heatmap regression and adopting direct keypoint regression. TokenPose [23] incorporated keypoint tokens into TransPose to enable the network to learn constraint information between keypoints. Within this framework, Dual-Pipeline Integrated Transformer (DPIT) [42] integrated top-down and bottom-up approaches through a parallel structure. In High-Resolution Transformer (HRFormer) [18], a novel backbone network was proposed, using pure Transformer to achieve the effect of HRNet [43].

The aforementioned methods all employ similar network architectures, as illustrated by the CNN+Transformer structure depicted in Fig. 2. Among them, TokenPose achieved optimal results, prompting us to adopt it as our baseline for experimentation. Of course, our approach can also be transferred to other token-based human pose estimation models.

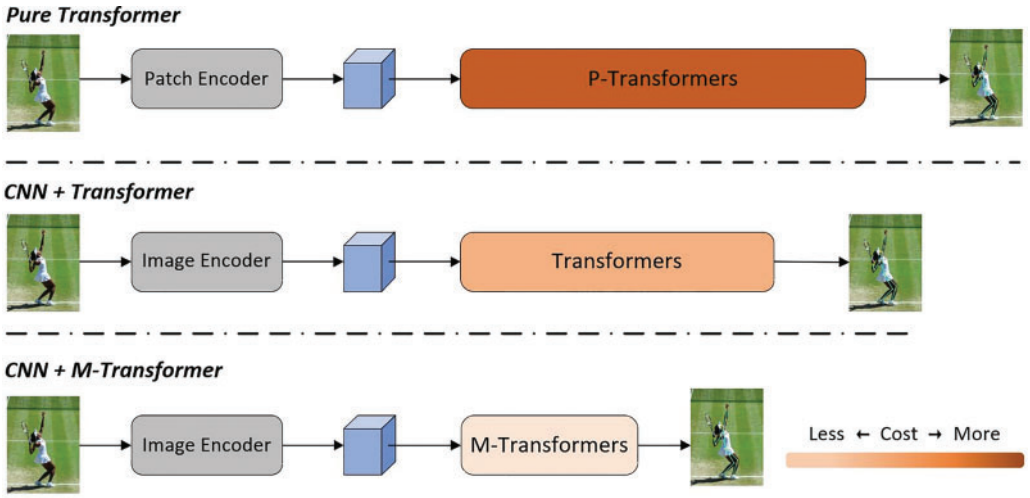

Figure 2: Complexity comparison between token-masked approach and pure transformer and CNN+Transformer architectures, with M-Transformer's M denoting mask

3 Proposed Method

Our goal is to propose a model for efficient human pose estimation. Firstly, we employ a human detector [11] to detect individuals in the input images. Then, the detected single-person images are fed into a backbone network [43] for feature extraction. The feature maps are converted into tokens and masked. The unmasked tokens, along with the keypoint tokens, are inputted into a Transformer for training. The output sequence is then remapped to a two-dimensional heatmap using Multilayer Perceptron (MLP) layer. Finally,

the heatmap is decoded into the coordinates of keypoints, which serve as the final predicted results. The overall architecture of TMPose is illustrated in Fig. 3. In the following sections, we will provide a detailed description of the proposed method.

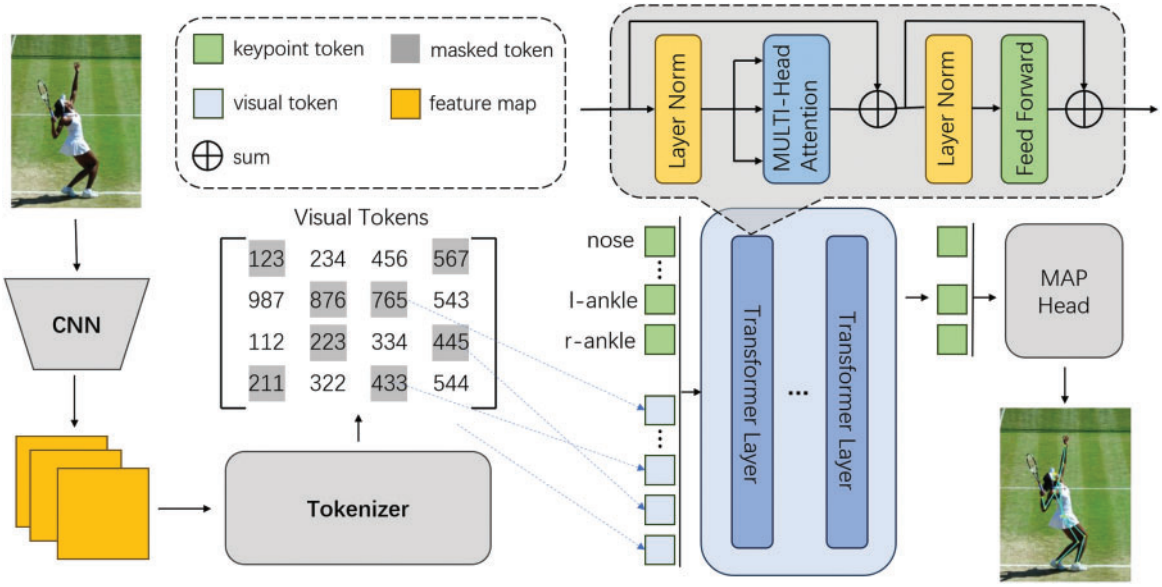

Figure 3: Overall framework of TMPose. The network use CNN for feature extraction, producing two-dimensional feature maps. The feature maps are then rearranged into one-dimensional token sequences in the Tokenizer. In the matrix of Visual Tokens, each element represents a token. Simultaneously, the network will randomly initialize and generate learnable keypoint tokens. Together with the visible tokens mentioned above, they are fed into the Transformer network for training. During the output stage, a multi-layer perceptron is employed to map the one-dimensional sequence back to a two-dimensional heatmap. Finally, the heatmap is decoded to obtain the predicted keypoints coordinates

3.1 CNN Backbone

In order to balance the cost of Transformer, we retained only the first three stages of HRNet [43] and named it HRNet-s, with a parameter size of only 25% of the original. Specifically, for a single-person image input, the network first crops it into a uniform size 256 × 192, followed by feature extraction through a deep convolutional neural network, resulting in a feature map of the original size 1/4. In TMPose, our visual tokens are obtained from the aforementioned feature maps rather than from image patches. We employ the CNNs to efficiently extract the low-level features of the images.

3.2 Feature Tokenizer

Transformer [44] is a sequence-to-sequence network, it is necessary to first map the two-dimensional feature maps into one-dimensional sequences. Following the approach in ViT [19], we assume that the feature map outputted by the backbone network is denoted as $x \in R^{H\times W\times C}$. We divide it into $\frac{H}{P_h} \times \frac{W}{P_w}$ grids and then flatten each grid into a one-dimensional sequence of size $P_h \times P_w \times C$. Subsequently, the rearranged sequences are mapped to the desired visual tokens using a linear layer. Simultaneously, the network initializes N learnable keypoint tokens, whose sequence length matches that of the visual tokens.

3.3 Token Masked

Considering that the concept of Masked Image Modeling has been thoroughly validated [24,25,33], we incorporate it into the pose transformer. The visual tokens be denoted as $v \in R^{L\times M}$ where L represents the sequence length and M denotes the sequence count. By employing a mapping function, we mask the visual tokens as $v' \in R^{L\times(M\cdot R)}$, $f: v \to v' \in R^{L\times(M\cdot R)}$. In this paper, we utilize three distinct mapping functions, namely random sampling, uniform sampling, and block sampling. Additionally, we adopt three different ratios denoted as R, which are 0.4, 0.6, and 0.8, respectively. We think that the choice of mask ratio is free. In order to improve the representativeness and reliability of our experiment, we selected three uniformly distributed values within this range −0.4, 0.6, and 0.8. The selection of this ratio is consistent with the commonly used settings in classical studies such as MAE, which has become a consensus.

Since Transformers lack the local feature extraction and layer-by-layer spatial awareness capabilities of convolutional neural networks, they are inherently insensitive to the positions of input features, making it challenging to capture spatial relationships between features directly. In human pose estimation, the relative positions of keypoints are tightly interrelated, requiring precise positional accuracy. Position encoding can aid the model in recognizing the relative locations of each keypoint, enhancing the clarity of the overall human structure. The two-dimensional positional embedding, denoted as pe_i [44], can provide positional annotations $\{visual\ \text{tokens}\} = \{v'_1 + pe_1, v'_2 + pe_2, \cdots, v'_H + pe_H\}$ for each input sequence, where $H = M \times R$ represents the number of sequences. Subsequently, the keypoint tokens are concatenated with the positional encoded visual tokens and fed into the transformer for training.

3.4 Transformer Module

We employed a multi-layer Transformer as the encoder, as illustrated in Fig. 3. Each Transformer layer primarily consists of multi-head self-attention and a feed-forward module. In addition, there are two normalization layers and a residual connection mechanism. Specifically, for the input sequence, we project it into three matrices of the same size through three linear mappings, generating Q (query), K (key), and V (value). Subsequently, these three matrices are fed into the multi-head self-attention module to compute attention scores:

$$MSA\left(Q, K, V\right) = soft\max\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) \cdot V \tag{1}$$

where d_k is the dimension of the key, and each score SA determines the attention level of the current query token.

3.5 Heatmap Generator

In the output stage of the Transformer, we select only N keypoint tokens for output. Subsequently, these tokens are remapped back to two-dimensional heatmaps form using a multi-layer perceptron. Specifically, $X \in R^{N\times H}$ is the output of the Transformer, where N represents the number of sequences and H denotes the sequence length. Then it passed into the multi-layer perceptron and generated $P \in R^{N\times H^*\times W^*}$. The H^* and W^* correspond to 1/4 of the original image size. Subsequently, P is reshaped into $P \in R^{N\times H\times W}$, resulting in a heatmap with the same dimensions as the original image. Finally, based on this heatmap, the coordinates of the keypoints on the human body are determined by locating the positions with the highest response. The specific results are illustrated in Fig. 4.

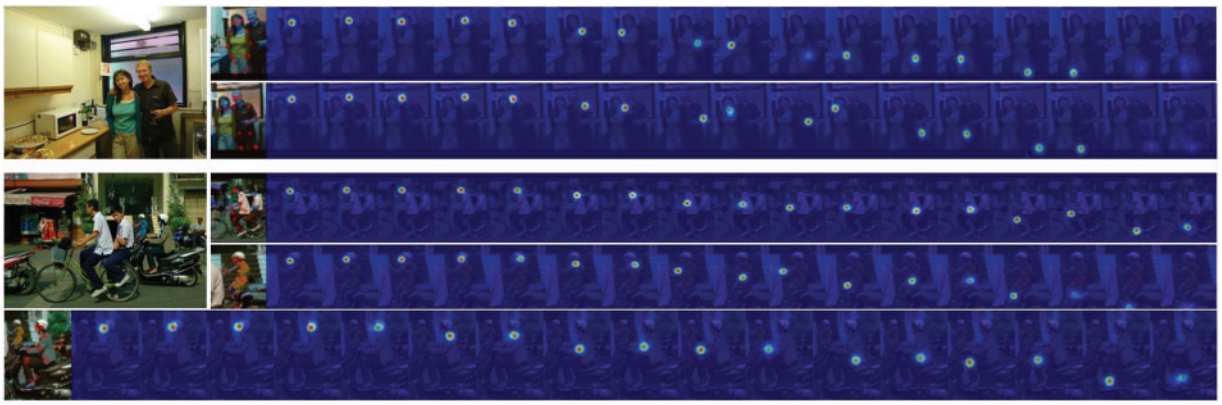

Figure 4: The network predicts the heatmap of persons, and each heatmap in each row represents the response to different joint points. It can be seen that although the persons have severe occlusion, the network can still distinguish different joint points well

4 Experiments

4.1 Experimental Details

In our experiments, we selected the COCO and MPII datasets, which are widely used in human pose estimation tasks and contain diverse poses and backgrounds, effectively supporting the training and validation of the model. For a fair comparison, we employed a model variant similar to TokenPose, as illustrated in Table 1. In the following text, we will use "B" to denote the Base model, "L" to denote the Large model, and "HRNet-s" to indicate the utilization of only the first three stages of HRNet.

Table 1: Configuration tables for different TMPoses, where GFLOPs are calculated using random sampling with a sampling rate of 60%

Model	CNN backbone	Layers	Heads	Patch size	#Params	GFLOPs
TMPose-Base	HRNetW32-s	12	8	4 × 3	13.5 M	4.9
TMPose-Large/D6	HRNetW48-s	6	8	4 × 3	20.8 M	9.2
TMPose-Large/D24	HRNetW48-s	24	12	4 × 3	27.5 M	9.4

In the training process, we followed a top-down paradigm. All single-person images in the COCO dataset were uniformly cropped to size 256 × 192, while in the MPII dataset, all single-person images were uniformly cropped to size 256 × 256, We utilized the Adam optimizer with an initial learning rate set at 1e − 3. At the 200th and 260th epochs, the learning rate was decreased to 1e − 4 and 1e − 5, respectively, with a total of 300 epochs for training. All models were implemented using the PyTorch framework and trained on a server equipped with 8 NVIDIA RTX 3090 24 G GPUs.

4.2 Dataset and Evaluation Metrics

We conducted experiments on two of the most widely used 2D human pose estimation datasets, COCO [45] and MPII [46]. The COCO dataset consists of 200 k images and 250 k human instances, with each instance annotated with 17 keypoints. For training, we utilized the COCO train2017 dataset, which contains 57 k images and 150 k human instances. The testing was performed on the COCO Validation

dataset, consisting of 5 k images. In the COCO dataset, the primary precision evaluation metrics are Average Precision (AP) and Average Recall (AR), which are calculated using the following formulas:

$$AP_{\mathrm{t}} = \frac{\sum_p \delta\left(OKS > \mathrm{t}\right)}{\sum_p 1} \tag{2}$$

where p is the number of detected human instances, and t is the threshold to refine the evaluation index. When t is taken as 0.5 and 0.75, it is noted as AP^{50} and AP^{75}, and similarly AR can be written as AR^{50}, AR^{75}. The evaluation metric commonly employed in the COCO dataset is the Object Keypoint Similarity (OKS), which is calculated using the following formula:

$$OKS = \frac{\sum_i \exp^{\left(-\frac{d_i^2}{2s^2k_i^2}\right)} \delta\left(v_i > 0\right)}{\sum_i \delta\left(v_i > 0\right)} \tag{3}$$

where d_{i} represents the Euclidean distance between detected keypoints and ground truth values, v_{i} indicates the visibility of ground truth values, s signifies the area of the human instance, and k_{i} represents the attenuation factor, δ denotes the normalization parameter of the keypoints. When $32^2 < s^2 < 96^2$ and $96^2 < s^2$, we write AP^{M} and AP^{L}, and similarly AR can be written as AR^{M}, AR^{L}. In the COCO dataset, the Object Keypoint Similarity (OKS) is ultimately transformed into the form of AP and AR. Specifically, we denote them as AP^{50} when OKS equals 0.5, AP^{75} when OKS equals 0.75. Additionally, for medium-sized objects, we have AP^{M}, while for large-sized objects, we use AP^{L}. Similarly, the AR is also divided based on the aforementioned criteria.

The MPII dataset comprises over 25 k images and 40 k human body instances. Each individual is annotated with 16 keypoints. The evaluation metric employed in the MPII dataset is the Percentage of Correct Keypoints (PCK), which is computed using the following formula:

$$PCKh = \sum_n \frac{||P - G||^2}{\alpha L^{head}} \cdot \frac{V_{(n\times 1)}}{c} \tag{4}$$

where P and G represent the predicted matrix and the ground truth matrix, respectively. c is the total number of visible landmarks, L^{head} is the length of the head, is a constant, and n denotes the batch size. Here, we adopt PCK@0.5(=0.5) as the final experimental result.

4.3 Quantitative Experimental

Table 2 presents the testing results of TMPose on the COCO validation dataset. TMPose achieves a significant reduction in computational complexity without a noticeable loss in accuracy. Specifically, TMPose-B exhibits a 60.5% decrease in computational complexity in the Transformer module, while experiencing a mere 0.3% decrease in AP. Similarly, TMPose-L/D24 demonstrates a 61.1% reduction in $GFLOPs^{\mathrm{T}}$ while only experiencing a 0.2% decrease in AP. In comparison to TransPose-H-A6, TMPose-L/D24 exhibits a substantial lead in both speed and accuracy. Furthermore, TMPose-L/D24 clearly outperforms Efficient human pose estimation network search framework (EfficientPose) and Lightweight High-Resolution Network (Lite-HRNet) in terms of both speed and accuracy.

Table 2: Test results on the COCO validation dataset, with an input image size of 256 × 192

Method	#Params	GFLOPs	$GFLOPs^T$	AP	AP^{50}	AP^{75}	AP^M	AP^L	AR
SimpleBaseline-R50 [11]	34.0 M	8.9	–	70.4	88.6	78.3	67.1	77.2	76.3
SimpleBaseline-R101 [11]	53.0 M	12.4	–	71.4	89.3	79.3	68.1	78.1	77.1
SimpleBaseline-R152 [11]	68.6 M	15.7	–	72.0	89.3	79.8	68.7	78.9	77.8
HRNet-W32 [43]	28.5 M	7.1	–	74.4	90.5	81.9	70.8	81.0	79.8
HRNet-W48 [43]	63.6 M	14.6	–	75.1	90.6	82.2	71.5	81.8	80.4
Lite-HRNet-18 [47]	1.1 M	0.2	–	64.8	86.7	73.0	62.1	70.5	71.2
Lite-HRNet-30 [47]	1.8 M	0.31	–	67.2	88.0	75.0	64.3	73.1	73.3
EfficientPose-B [48]	3.3 M	1.1	–	71.1	–	–	–	–	–
EfficientPose-C [48]	5.0 M	1.6	–	71.3	–	–	–	–	–
TransPose-R-A4 [37]	6.0 M	8.9	3.38	72.6	89.1	79.9	68.8	79.8	78.0
TransPose-H-S [37]	8.0 M	10.2	4.88	74.2	89.6	80.8	70.6	81.0	79.5
TransPose-H-A6 [37]	17.5 M	21.8	11.4	75.8	90.1	82.1	71.9	82.8	80.8
TokenPose-B* [23]	13.5 M	5.7	1.29	75.6	92.6	82.7	72.8	80.1	78.4
TokenPose-L/D6* [23]	20.8 M	10.3	1.93	76.7	92.6	83.1	74.0	81.1	79.4
TokenPose-L/D24* [23]	27.5 M	11.0	2.57	76.9	92.6	83.7	74.2	81.3	79.6
TMPose-B* (ours)	13.5 M	4.9 (−14%)	0.51 (−60.5%)	75.3 (−0.3%)	92.6	82.6	72.3	79.7	78.0
TMPose-L/D6* (ours)	20.8 M	9.2 (−11%)	0.75 (−61.1%)	76.3 (−0.4%)	92.5	82.7	73.6	81.0	79.2
TMPose-L/D24* (ours)	27.5 M	9.4 (−15%)	1.00 (−61.1%)	76.7 (−0.2%)	92.6	83.7	73.9	81.0	79.2

Note: The asterisk * indicates the usage of GTBox, and $GFLOPs^T$ represents the computational complexity of the Transformer component. Due to TMPose's focus on accelerating the Transformer, we primarily evaluate its performance based on $GFLOPs^T$, with AP (Average Precision) as the primary metric for assessing accuracy.

Table 3 presents the testing results of TMPose on the MPII dataset. TMPose-L/D6 achieves a significant reduction in computational complexity, with a decrease of 59.4%. Meanwhile, the Mean metric only experiences a marginal loss of 0.3%. Specifically, TMPose-L/D6 performs on par with TokenPose-L/D6 in terms of accuracy for the shoulder (Sho) joint and even surpasses it for the knee (Kne) joint, achieving an impressive precision of 86.2%. Therefore, it can be inferred that the token masking approach does not compromise the overall accuracy. Instead, it allows for a more efficient pose estimation network, striking a balance between speed and precision.

Table 3: Test results on MPII dataset, with an input image size of 256 × 256

Method	#Params	$GFLOPs^T$	Head	Sho	Elb	Wri	Hip	Kne	Ank	Mean
SimpleBaseline-R50 [11]	34.0 M	–	96.4	95.3	89.0	83.2	88.4	84.0	79.6	88.5
SimpleBaseline-R101 [11]	53.0 M	–	96.9	95.9	89.5	84.4	88.4	84.5	80.7	89.1
SimpleBaseline-R152 [11]	53.0 M	–	97.0	95.9	90.0	85.0	89.2	85.3	81.3	89.6
HRNetW32 [43]	28.5 M	–	96.9	96.0	90.6	85.8	88.7	86.6	82.6	90.1
TokenPose-L/D6* [23]	21.4 M	0.64	97.1	95.9	91.0	85.8	89.5	86.1	82.7	90.2
TMPose-L/D6*	21.4 M	0.26 (−59.4%)	96.6	95.9	90.1	85.7	89.2	86.2	82.3	89.9 (−0.3%)

Note: The asterisk * indicates the usage of GTBox, Mean is when PCK@0.5 and GFLOPsT represent the computational complexity of the Transformer component.

4.4 Experimental Details

As shown in Fig. 5, it can be observed that TMPose can effectively predict the final keypoints even in the presence of masks. It demonstrates excellent performance in both single-person and multi-person scenarios, as well as in crowded or occluded scenes. Fig. 6 presents a comparison of experimental results between TMPose and mainstream lightweight network models. From the first two images, TMPose can infer

more reasonable keypoint positions in occluded scenes. Moreover, the fourth image reveals that TMPose can achieve more accurate predictions in scenes with complex image features. In conclusion, TMPose outperforms EfficientPose in terms of both speed and accuracy.

Figure 5: Visualization prediction results of TMPose on the COCO validation dataset

4.5 Ablation Experiment

To verify the effects of different mask types and sampling rates on the experiment, we conducted ablation experiments as shown in Table 4. It can be observed that under the same sampling rate, random sampling achieved the best accuracy, followed by uniform sampling, while block sampling resulted in the highest performance loss.

In this experiment, we conducted comprehensive testing on three masking methods (random, uniform, and block) and three masking rates, generating nine different sets of data parameters. The results showed that a random mask with a mask rate of 0.6 performed the best, while the effects of uniform masks and block masks were relatively low under the same mask rate. In the case of the same mask type, as the mask rate decreases, the efficiency increases accordingly. The specific reasons are explained as follows:

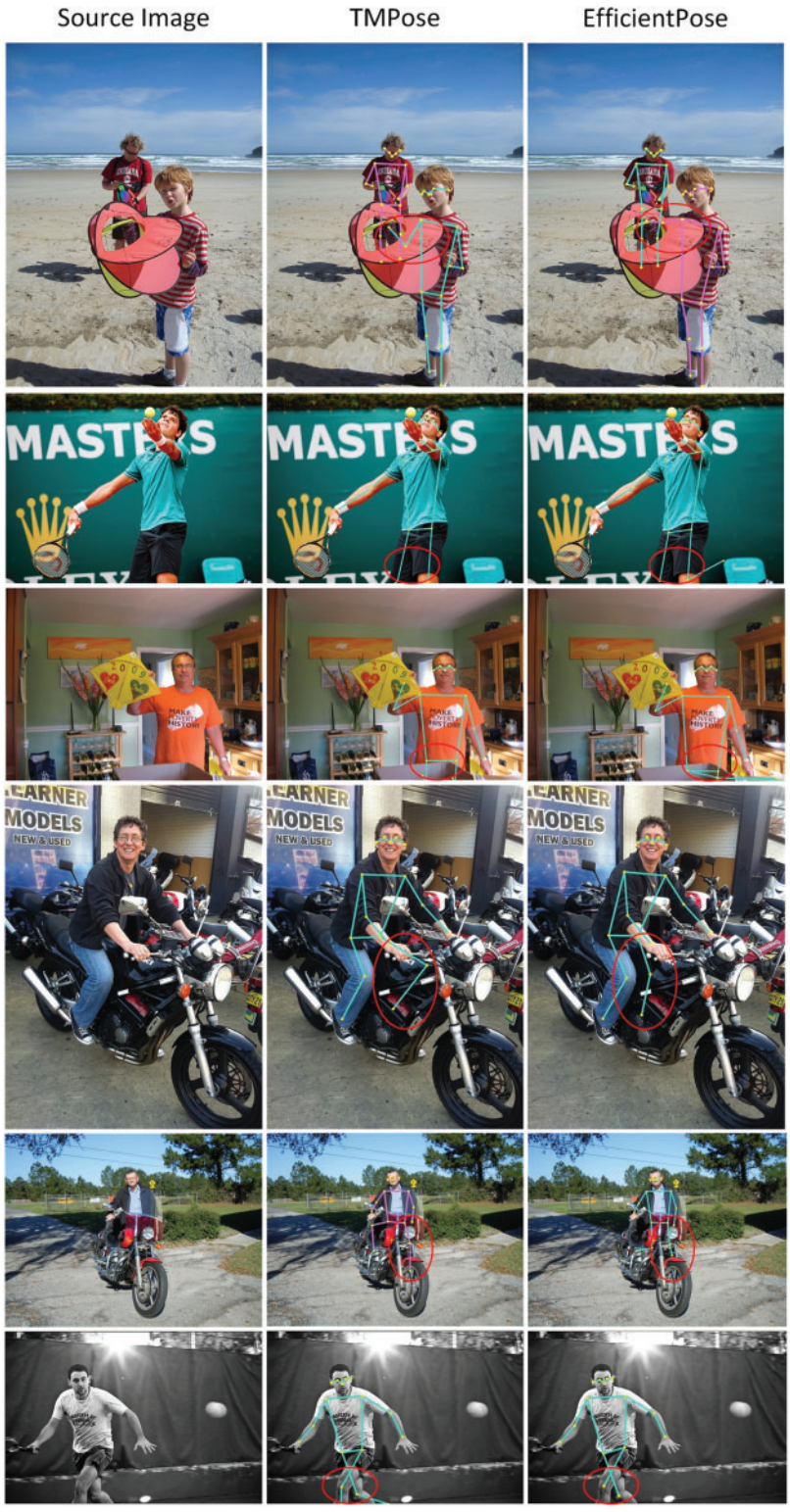

Figure 6: Comparison results between TMPose and other efficient model on the COCO validation dataset

Table 4: Experimental results of TMPose-B with different sampling rates and mask types on COCO validation dataset

Method	Mask ratio	Mask type	GFLOPs	GFLOPs^T	AP	AR
TokenPose-B [23]	1	None	5.7	1.29	75.6	78.4
		Random			75.6	78.4
TMPose-B	0.8	Uniform	5.2 (−9%)	0.85 (−34%)	75.6	78.3
		Block			75.5	78.2
		Random			75.3	78.0
TMPose-B	0.6	Uniform	4.9 (−14%)	0.51 (−60%)	75.2	77.9
		Block			75.1	77.8
		Random			75.0	77.8
TMPose-B	0.4	Uniform	4.6 (−19%)	0.26 (−79%)	74.9	77.7
		Block			74.6	77.4

COCO is a discrete dataset, and the data is randomly distributed. Using a random mask is more in line with the statistical rules of the COCO dataset. Uniform masks and block masks, due to their fixed nature, may lead to the neglect or loss of some important features. Uniform masks are relatively uniform in feature selection, although they can maintain a certain amount of information, fixed selection methods may not be able to adapt to changes in feature distribution. The block mask more clearly restricts the selection of information, which may lead to the destruction of the correlation between adjacent features and affect the model's judgment ability. This is contrary to the random distribution of the dataset and can have a certain impact on performance.

Secondly, regarding the impact of mask rate, a high mask rate can theoretically improve the model's coverage and accuracy of data, but it is also affected by background noise, which can affect recognition accuracy and inevitably lead to high computational complexity. However, in practical applications, a relatively low mask rate can enable the model to retain feature information while removing redundant information, thereby improving computational efficiency. A low mask rate may result in the loss of relevant useful information, affecting accuracy. Experiments have shown that, As the sampling rate gradually decreased, the computational complexity also decreased progressively. Specifically, when the sampling rate was 0.4, GFLOPs^T decreased by 79%, and when the sampling rate was 0.6, GFLOPs^T decreased by 60%.

Based on the above analysis, the author believes that Random 0.6 achieves a balance between speed and accuracy, making it the optimal combination. Therefore, when selecting masking strategies and mask rates, it is crucial to balance accuracy and efficiency to achieve optimal performance with available computational resources.

5 Conclusion

In this paper, we introduce a Token Masked-based human pose estimation network called TMPose and demonstrate its effectiveness on the COCO and MPII datasets. Our method achieves significant computational efficiency, reducing GFLOPs while maintaining competitive accuracy comparable to TokenPose. The AP (Average Precision) metric on the COCO validation dataset confirms that TMPose accurately predicts human keypoint coordinates with reduced computational overhead, making it highly suitable for practical applications. Qualitative experiments further validate TMPose's superior performance over other lightweight models in real-world scenarios. Additionally, our ablation experiments provide insight into the influence of different Mask Ratios and Mask Types on TMPose's accuracy, revealing optimal configurations. These findings contribute to the state-of-the-art by offering a pathway to lightweight, accurate pose estimation for human keypoint detection tasks.

However, we observed certain limitations in Token Masking when applied to pose estimation networks with shallow or non-CNN backbones. When using Stem Net or patch-based tokenization without a deep CNN backbone, the method experiences slow convergence and accuracy degradation. For example, recent studies, such as those in Fine-Grained Structure Aggregation Network (FSA-Net) [49], Channel Spatial Integrated Transformer (CSIT) [9], Efficient Posenet with Coarse to Fine Transformer (CFPose) [50], and other related works, have also explored this issue. We will take these studies into account as we plan our next steps, with the aim of extending TMPose's applicability across various architectures and further advancing the field.

Acknowledgement: The authors would like to acknowledge the server support provided by Associate Professor Haixiang Zhang and the Scientific Research Start-Up Fund of Zhejiang Sci-Tech Funding from the University.

Funding Statement: This work is supported in part by the Scientific Research Start-Up Fund of Zhejiang Sci-Tech University, under the project titled "(National Treasury) Development of a Digital Silk Museum System Based on Metaverse and AR" (Project No. 11121731282202-01).

Author Contributions: Conceptualization, Xinyi Song; methodology, Xinyi Song and Haixiang Zhang; software, Xinyi Song; validation, Haixiang Zhang, Xinyi Song, Shaohua Li; formal analysis, Xinyi Song; investigation, Shaohua Li; resources, Haixiang Zhang; data curation, Xinyi Song; writing—original draft preparation, Shaohua Li; writing—review and editing, Xinyi Song, Haixiang Zhang; visualization, Shaohua Li; supervision, Haixiang Zhang; project administration, Haixiang Zhang; funding acquisition, Haixiang Zhang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available at https://github.com/lshua98/tmpose (accessed on 9 January 2025).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Geng Z, Wang C, Wei Y, Liu Z, Li H, Hu H. Human pose as compositional tokens. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–24; Vancouver, BC, Canada: IEEE; 2023. p. 660–71. doi:10.1109/CVPR52729.2023.00071
2. Wang Y, Xia Y, Liu S. BCCLR: a skeleton-based action recognition with graph convolutional network combining behavior dependence and context clues. Comput Mater Contin. 2024;78(3):4489–507. doi:10.32604/cmc.2024.048813.
3. Cao Z, Simon T, Wei SE, Sheikh Y. Realtime multi-person 2D pose estimation using part affinity fields. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA: IEEE; 2017. p. 1302–10. doi:10.1109/CVPR.2017.143.
4. Liu L, Sun Y, Ge X. A hybrid multi-person fall detection scheme based on optimized YOLO and ST-GCN. Int J Interact Multimed Artif Intell. 2024. doi:10.9781/ijimai.2024.09.003.
5. Salisu S, Mohamed ASA, Jaafar MH, Pauzi ASB, Younis HA. A survey on deep learning-based 2D human pose estimation models. Comput Mater Contin. 2023;76(2):2385–400. doi:10.32604/cmc.2023.035904.
6. Ishwarya K, Alice Nithya A. Squirrel search optimization with deep convolutional neural network for human pose estimation. Comput Mater Contin. 2023;74(3):6081–99. doi:10.32604/cmc.2023.034654.
7. Toshev A, Szegedy C. DeepPose: human pose estimation via deep neural networks. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition; 2014 Jun 23–28; Columbus, OH, USA: IEEE; 2014. p. 1653–60. doi:10.1109/CVPR.2014.214

8. Wei SE, Ramakrishna V, Kanade T, Sheikh Y. Convolutional pose machines. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Jun 27–30; Las Vegas, NV, USA: IEEE; 2016. p. 4724–32. doi:10.1109/CVPR.2016.511.
9. Li S, Zhang H, Ma H, Feng J, Jiang M. CSIT: channel spatial integrated transformer for human pose estimation. IET Image Process. 2023;17(10):3002–11. doi:10.1049/ipr2.12850.
10. Papandreou G, Zhu T, Kanazawa N, Toshev A, Tompson J, Bregler C, et al. Towards accurate multi-person pose estimation in the wild. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA: IEEE; 2017. p. 3711–9. doi:10.1109/CVPR.2017.395.
11. Xiao B, Wu H, Wei Y. Simple baselines for human pose estimation and tracking. In: Computer vision–ECCV 2018. Cham: Springer International Publishing; 2018. p. 472–87. doi:10.1007/978-3-030-01231-1_29.
12. Liu Z, Lin W, Shi Y, Zhao J. A robustly optimized BERT pre-training approach with post-training. In: Chinese computational linguistics. Cham: Springer International Publishing; 2021. p. 471–84. doi:10.1007/978-3-030-84186-7_31.
13. Radford A, Narasimhan K, Salimans T, Sutskever I. Improving language understanding by generative pre-training. 2018 [cited 2024 Dec 10]. Available from: https://www.mikecaptain.com/resources/pdf/GPT-1.pdf.
14. Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, et al. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv:1910.13461. 2019.
15. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res. 2020;21(1):5485–551.
16. Zhang S, Roller S, Goyal N, Artetxe M, Chen M, Chen S, et al. OPT: open pre-trained transformer language models. arXiv:2205.01068. 2022.
17. Yuan Y, Fu R, Huang L, Lin W, Zhang C, Chen X, et al. HRFormer: high-resolution vision transformer for dense predict. Adv Neural Inf Process Syst. 2021;34:7281–93.
18. Xu Y, Zhang J, Zhang Q, Tao D. ViTPose: simple vision transformer baselines for human pose estimation. arXiv:2204.12484. 2022.
19. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16 × 16 words: transformers for image recognition at scale. arXiv:2010.11929. 2020.
20. Han K, Xiao A, Wu E, Guo J, Xu C, Wang Y. Transformer in transformer. arXiv:2103.00112. 2021.
21. Zheng S, Lu J, Zhao H, Zhu X, Luo Z, Wang Y, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA: IEEE; 2021. p. 6881–90.
22. Zhu X, Su W, Lu L, Li B, Wang X, Dai J. Deformable detr: deformable transformers for end-to-end object detection. arXiv:2010.04159. 2020.
23. Li Y, Zhang S, Wang Z, Yang S, Yang W, Xia ST, et al. TokenPose: learning keypoint tokens for human pose estimation. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10–17; Montreal, QC, Canada: IEEE; 2021. p. 11293–302. doi:10.1109/ICCV48922.2021.01112.
24. He K, Chen X, Xie S, Li Y, Dollár P, Girshick R. Masked autoencoders are scalable vision learners. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA: IEEE; 2022. p. 15979–88. doi:10.1109/CVPR52688.2022.01553.
25. Bao H, Dong L, Piao S, Wei F. Beit: bert pre-training of image transformers. arXiv:2106.08254. 2021.
26. Wang Y, Xu Z, Wang X, Shen C, Cheng B, Shen H, et al. End-to-end video instance segmentation with transformers. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA: IEEE; 2021. p. 8741–50.
27. Han S, Mao H, Dally WJ. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv: 1510.00149. 2015.
28. Chen T, Cheng Y, Gan Z, Yuan L, Zhang L, Wang Z. Chasing sparsity in vision transformers: an end-to-end exploration. arXiv:2106.04533. 2021.
29. Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv:1503.02531. 2015.

30. Chen X, Cao Q, Zhong Y, Zhang J, Gao S, Tao D. DearKD: data-efficient early knowledge distillation for vision transformers. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA: IEEE; 2022. p. 12042–52. doi:10.1109/CVPR52688.2022.01174.
31. Shen S, Dong Z, Ye J, Ma L, Yao Z, Gholami A, et al. Q-BERT: hessian based ultra low precision quantization of BERT. Proc AAAI Conf Artif Intell. 2020;34(5):8815–21. doi:10.1609/aaai.v34i05.6409.
32. Sun M, Ma H, Kang G, Jiang Y, Chen T, Ma X, et al. VAQF: fully automatic software-hardware co-design framework for low-bit vision transformer. arXiv:2201.06618. 2022.
33. Li Y, Fan H, Hu R, Feichtenhofer C, He K. Scaling language-image pre-training via masking. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–24; Vancouver, BC, Canada: IEEE; 2023. p. 23390–400. doi:10.1109/CVPR52729.2023.02240.
34. Yuan L, Chen Y, Wang T, Yu W, Shi Y, Jiang Z, et al. Tokens-to-Token ViT: training vision transformers from scratch on ImageNet. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10–17; Montreal, QC, Canada: IEEE; 2021. p. 538–47. doi:10.1109/ICCV48922.2021.00060.
35. Rao Y, Zhao W, Liu B, Lu J, Zhou J, Hsieh CJ. DynamicViT: efficient vision transformers with dynamic token sparsification. arXiv:2106.02034. 2021.
36. Ma H, Wang Z, Chen Y, Kong D, Chen L, Liu X, et al. PPT: token-pruned pose transformer for monocular and multi-view human pose estimation. In: Computer vision-ECCV 2022. Cham: Springer Nature Switzerland; 2022. p. 424–42. doi:10.1007/978-3-031-20065-6_25.
37. Yang S, Quan Z, Nie M, Yang W. TransPose: keypoint localization via transformer. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2017 Oct 10–17; Montreal, QC, Canada: IEEE; 2017. p. 11782–92. doi:10.1109/ICCV48922.2021.01159.
38. Mao W, Ge Y, Shen C, Tian Z, Wang X, Wang Z. TFPose: direct human pose estimation with transformers. arXiv:2103.15320. 2021.
39. Li K, Wang S, Zhang X, Xu Y, Xu W, Tu Z. Pose recognition with cascade transformers. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA: IEEE; 2021. p. 1944–53. doi:10.1109/CVPR46437.2021.00198.
40. Lin K, Wang L, Liu Z. End-to-end human pose and mesh reconstruction with transformers. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA: IEEE; 2021. p. 1954–63. doi:10.1109/CVPR46437.2021.00199.
41. Zheng C, Zhu S, Mendieta M, Yang T, Chen C, Ding Z. 3D human pose estimation with spatial and temporal transformers. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10–17; Montreal, QC, Canada: IEEE; 2021. p. 11636–45. doi:10.1109/ICCV48922.2021.01145.
42. Zhao S, Liu K, Huang Y, Bao Q, Zeng D, Liu W. DPIT: dual-pipeline integrated transformer for human pose estimation. In: Artificial intelligence. Cham: Springer Nature Switzerland; 2022. p. 559–76. doi:10.1007/978-3-031-20500-2_46.
43. Sun K, Xiao B, Liu D, Wang J. Deep high-resolution representation learning for human pose estimation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019 Jun 15–20; Long Beach, CA, USA: IEEE; 2019. p. 5693–5703.
44. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. arXiv:1706.03762. 2017.
45. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference ; 2014 Sep 6–12; Zurich, Switzerland: Springer International Publishing; 2014. p. 740–55.
46. Andriluka M, Pishchulin L, Gehler P, Schiele B. 2D human pose estimation: new benchmark and state of the art analysis. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition; 2014 Jun 23–28; Columbus, OH, USA: IEEE; 2014. p. 3686–93. doi:10.1109/CVPR.2014.471.
47. Yu C, Xiao B, Gao C, Yuan L, Zhang L, Sang N, et al. Lite-HRNet: a lightweight high-resolution network. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA: IEEE; 2021. p. 10440–50.

48. Zhang W, Fang J, Wang X, Liu W. EfficientPose: efficient human pose estimation with neural architecture search. Comput Vis Medium. 2021;7(3):335–47. doi:10.1007/s41095-021-0214-z.
49. Li S, Zhang H, Ma H, Feng J, Jiang M. Full scale-aware balanced high-resolution network for multi-person pose estimation. Comput Mater Contin. 2023;76(3):3379–92. doi:10.32604/cmc.2023.041538.
50. Li S, Zhang H, Ma H, Feng J, Jiang M. Efficient posenet with coarse to fine transformer. In: ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2024 Apr 14–19; Seoul, Republic of Korea: IEEE; 2024. p. 5100–4. doi:10.1109/ICASSP48485.2024.10448008.