

GTPT: Group-based Token Pruning Transformer for Efficient Human Pose Estimation

Haonan Wang^{1,2*}, Jie Liu¹, Jie Tang¹, Gangshan Wu¹, Bo Xu²,
Yanbing Chou², and Yong Wang²

¹ State Key Laboratory for Novel Software Technology, Nanjing University, China
² Cainiao Network, China

wanghaonan0522@gmail.com, {liujie,tangjie,gswu}@nju.edu.cn,
{songbai.xb,tonychou.zyb,Richard.wangy}@cainiao.com
<https://github.com/haonanwang0522/GTPT>

Abstract. In recent years, 2D human pose estimation has made significant progress on public benchmarks. However, many of these approaches face challenges of less applicability in the industrial community due to the large number of parametric quantities and computational overhead. Efficient human pose estimation remains a hurdle, especially for whole-body pose estimation with numerous keypoints. While most current methods for efficient human pose estimation primarily rely on CNNs, we propose the Group-based Token Pruning Transformer (GTPT) that fully harnesses the advantages of the Transformer. GTPT alleviates the computational burden by gradually introducing keypoints in a coarse-to-fine manner. It minimizes the computation overhead while ensuring high performance. Besides, GTPT groups keypoint tokens and prunes visual tokens to improve model performance while reducing redundancy. We propose the Multi-Head Group Attention (MHGA) between different groups to achieve global interaction with little computational overhead. We conducted experiments on COCO and COCO-WholeBody. Compared to other methods, the experimental results show that GTPT can achieve higher performance with less computation, especially in whole-body with numerous keypoints.

Keywords: Efficient human pose estimation · Whole-body pose estimation · Transformer · Token pruning · Group

1 Introduction

2D human pose estimation (HPE) [4, 22, 32, 39] is a core computer vision task that localizes human anatomical joints in images, underpinning applications such as 3D pose estimation [8, 36, 47, 51], activity recognition [31, 41], and pose tracking [33]. It is a field of significant interest in both academic and industrial sectors.

Despite achieving superior performance, 2D HPE still faces challenges in computation efficiency, particularly for whole-body pose estimation. Previous works

*: This work was conducted during Haonan Wang’s internship at Cainiao Network.

✉: Corresponding author. liujie@nju.edu.cn

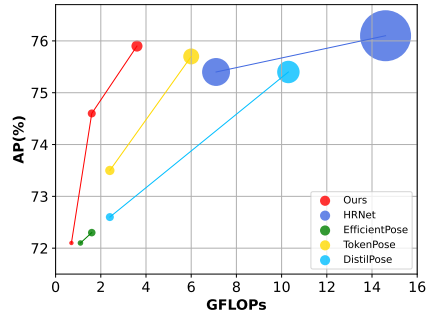


Fig. 1: Comparison of our method with SOTA methods on COCO val. The horizontal coordinate indicates computation, the vertical coordinate indicates precision, and the circle size indicates the model parameters.

on efficient pose estimation relied heavily on CNN-based methods to preserve the 2D structure. However, when the number of keypoints rises dramatically, high-resolution feature maps struggle to distinguish between dense keypoints, such as those on the face. Besides, keypoints are highly correlated and strongly constrained by dynamics and physics [28]. Unfortunately, CNN-based methods often struggle to capture the long-distance correlation between individual keypoints due to the limited receptive field.

In recent years, Transformer has shown tremendous potential in various vision tasks, including pose estimation. TokenPose [15] has effectively demonstrated the power of Transformer, as the attention mechanism can explicitly model relationships between keypoints. It conducts two tokenizations: keypoint tokens representing keypoint objects and visual tokens representing uniformly split image patches. Meanwhile, the Transformer can dynamically adjust the weights and accept variable-length inputs, resulting in fewer parameters and computational effort. Hence, we aim to leverage the superiority of the Transformer to develop an efficient pose estimation method. The previous method involves encoding all keypoints into keypoint tokens, which are concatenated with visual tokens and fed into the Transformer for feature extraction. However, this approach introduces redundancy that negatively impacts efficiency. We investigate the redundancy of the model from two perspectives: keypoint tokens and visual tokens. For keypoint tokens, redundancy increases significantly as the number of keypoints rises. It is particularly evident in body parts with a high density of keypoints. In shallow layers, keypoints within the same body part often focus on similar areas. Therefore, individually modeling each keypoint becomes highly redundant, especially when dealing with numerous keypoints. Besides, numerous visual tokens are redundant. In shallow layers, the attention of each keypoint token should primarily focus on the estimated human body rather than the background. As the model goes deeper, each keypoint token only focuses on a specific area around the keypoint, rather than the entire human body.

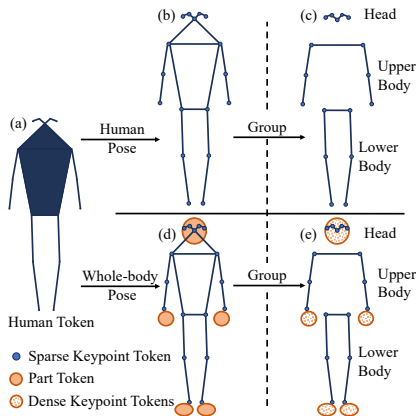


Fig. 2: Overview of the introduction and grouping of keypoints. GTPT introduces keypoints in a coarse-to-fine manner. It starts with a human token and gradually transitions to sparse keypoint tokens and part tokens. Eventually, it converts part tokens into corresponding dense keypoint tokens. Besides, we categorize all keypoints into three groups: head, upper body, and lower body.

Therefore, we propose a Group-based Token Pruning Transformer (GTPT) to enhance performance while minimizing redundancy. To mitigate the impact of numerous keypoints on efficiency without compromising performance, we propose a stepwise strategy to refine keypoints in a coarse-to-fine manner. Firstly, we categorize all the keypoints into two types: sparse keypoints on the body, and dense keypoints located elsewhere. As illustrated in Fig. 2 (a), only one human token is input initially. Subsequently, the human token gradually transforms into sparse keypoint tokens. For whole-body pose estimation, we introduce five part tokens (for the head, both hands, and both feet, as shown in Fig. 2 (d)) to be converted into dense keypoint tokens in the later stage. Besides, we employ pruning visual tokens to minimize redundancy. However, directly pruning with a high rate often prioritizes easily localizable keypoints while neglecting difficult-to-localize ones. To preserve visual tokens associated with keypoints, we propose group-based pruning, which enhances the model’s performance and robustness to pruning. As depicted in Fig. 2 (c) and (e), we categorize keypoints into three groups: head, upper body, and lower body. To enhance performance, we model each group separately with masks. Furthermore, simply modeling keypoints within a group is insufficient for constraining the entire human body. Therefore, we propose the Multi-Head Group-Attention (MHGA), which not only models the keypoints within each group but also captures relationships among all keypoints while maintaining a low computational overhead.

Compared with previous works, our design offers three merits. (1) We boost performance with reduced computational overhead by group-based token pruning, as shown in Fig. 1. (2) GTPT introduces keypoints incrementally, following

a coarse-to-fine strategy, thereby minimizing the impact of their increasing number on computation efficiency. (3) Our method intelligently prunes more visual tokens for areas with a higher density of keypoints, such as the face, while retaining more visual tokens for regions with fewer keypoints but more coverage, like the lower body, resulting in enhanced efficiency.

The contributions are summarized as follows:

- We propose a novel Group-based Token Pruning Transformer (GTPT) for efficient human pose estimation, especially whole-body pose estimation, which improves performance while reducing redundancy by grouping keypoint tokens and pruning visual tokens.
- We propose Multi-Head Group-Attention (MHGA) to model the keypoints within each group and capture the overall relationship between all the keypoints while keeping the computational overhead low.
- We propose a coarse-to-fine strategy to incrementally introduce keypoints, effectively mitigating the impact of numerous keypoints on efficiency.

2 Related Work

2.1 Transformer in HPE

Transformer [30] has not only achieved remarkable success in Natural Language Processing (NLP) but has also shown strong performance in computer vision, including classification [6, 18], detection [50], and segmentation [5]. Thus, pose estimation has also witnessed advancements, transitioning from CNNs to Transformers. We can broadly categorize previous works utilizing Transformers in HPE into two types: refining image features and aggregating part-wise features.

Refining image features. This manner enhances the feature extraction for images. For instance, TransPose [42] combines transformer encoder layers and CNN to capture global relationships effectively. HRFormer [46] replaces CNN with transformer encoders in HRNet [27] to extract superior high-resolution features. TCFormer [48] employs progressive clustering to merge tokens and preserve diverse scale information. ViTPose [39] leverages a pre-trained SOTA transformer backbone to enhance accuracy.

Aggregating part-wise features. Each keypoint is abstracted as a token to aggregate keypoint features. TFPose [21] enhances CNN features with transformer encoders. Transformer decoders are then employed to extract keypoints features for regressing coordinates. Poseur [22] adopts deformable cross-attention to extract keypoints features. TokenPose [15] feeds keypoint and visual tokens to the transformer for feature extraction, followed by predicting heatmaps using keypoint tokens. We incorporate transformers to query visual cues and learn anatomical constraints through attention. However, our approach differs from previous approaches in introducing keypoints in a coarse-to-fine manner. Furthermore, we strategically group keypoints to minimize computational costs while ensuring consistent performance, particularly in scenarios involving numerous keypoints, such as whole-body situations.

2.2 Efficient HPE

To improve efficiency, current work for efficient pose estimation concentrates on three key aspects: efficient architecture design, distillation, and pruning.

Efficient architecture design. Many works focus on designing efficient architectures for this task [1, 23, 25, 26, 35, 45]. EfficientPose [1] adopts neural architecture search (NAS) to obtain an efficient backbone network. Besides, ZoomNAS [38] also employs a NAS framework to jointly search for connections between model architectures and different sub-modules, aiming to improve accuracy and efficiency. Lite-HRNet [45] introduces shuffle blocks to HRNet and incorporates an efficient conditional channel weighting unit to replace the costly 1×1 convolution in shuffle blocks. RTMPose [12] investigates factors influencing the performance and latency of pose estimation and develops a real-time model.

Distillation. Knowledge distillation (KD) [10] aims to transfer knowledge from teacher to student. FDP [49] firstly introduces KD into pose estimation. OKDHP [16] proposes an online distillation approach that transfers pose structure knowledge in a one-stage manner. DistilPose [44] introduces a heatmap-to-regression distillation framework that combines both benefits. DWPose [43] proposes a two-stage distillation to enhance the performance of whole-body pose estimation.

Pruning. The purpose of pruning is to reduce redundant visual tokens. PPT [19] proposes to localize the human body region and prune the background region by summing the attention maps of keypoint tokens to obtain a score. Besides, PPT adopts a pruning strategy that gradually increases the pruning ratio during training, which may negatively affect performance when the pruning ratio becomes too large. Therefore, we utilize a global perceived loss to transfer knowledge from the unpruned model to the pruned model. Besides, pruning may prune visual tokens associated with challenging keypoints, as it tends to focus on easily recognizable ones. Therefore, we introduce a grouping-based pruning method to retain visual tokens corresponding to all keypoints as much as possible.

3 Method

The key idea of our method is to take full advantage of the Transformer and improve computational efficiency by minimizing redundancy. Thus, we propose a Group-based Token Pruning Transformer (GTPT). Fig. 3 illustrates the architecture, comprising the Tokenizer, Coarse Encoder, Coarse-to-fine Module, and Fine Encoder. GTPT effectively reduces redundancy by introducing keypoints in a coarse-to-fine manner and employing group-based pruning for visual tokens.

3.1 Tokenizer

To meet the input requirements of Transformer [6, 30], a tokenizer is necessary to convert the 2D image \mathbf{I} into a 1D sequence. Following TokenPose [15], the input image \mathbf{I} is passed through a shallow CNN backbone to obtain the feature map $\mathbf{F} \in \mathbb{R}^{H \times W \times C}$, where H , W , and C denote the height, width, and channel

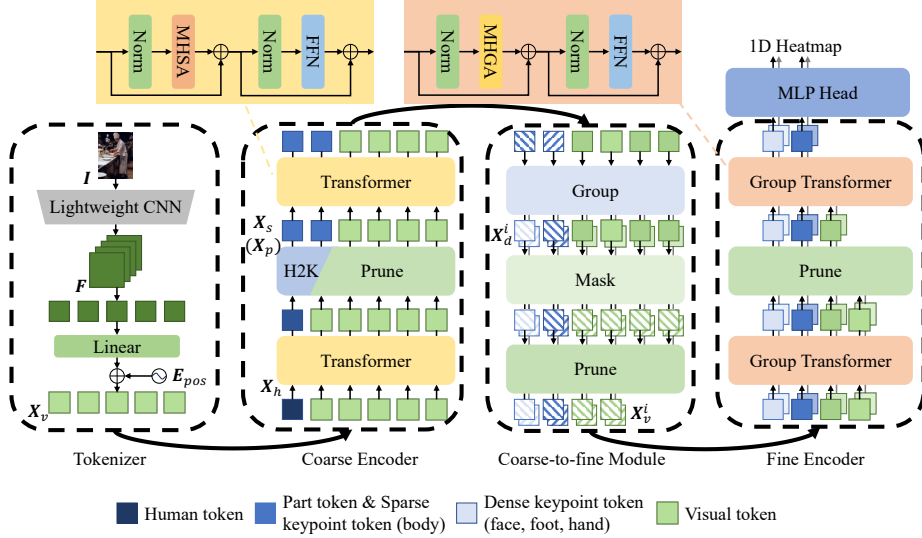


Fig. 3: Overview of our proposed architecture. First, the tokenizer, which consists of a shallow CNN, performs feature extraction on the input image and transforms it into a sequence of tokens. Next, the Coarse Encoder, made of Transformer layers, gradually performs feature extraction from the target human to sparse keypoints as the network depth increases. The Coarse-to-Fine Module is the transition between the Coarse Encoder and the Fine Encoder, which introduces dense keypoints and groups the keypoints. To extract better features for each group, it masks the visual tokens differently in different groups. We then perform feature extraction on the keypoints of each group using the Fine Encoder. Finally, all keypoint tokens are fed into the unified MLP Head to estimate the 1D heatmaps of each keypoint.

number. For more efficiency, different from TokenPose, we utilize ShuffleNet V2 [20] as the CNN backbone. The feature map F is then uniformly divided into $N_{vis} = \frac{H}{P_H} \times \frac{W}{P_W}$ non-overlapping patches, where (P_H, P_W) denotes the resolution of each patch. Each patch is flattened into a 1D vector of size $C \times P_H \times P_W$ and fed into a linear projection function to obtain the visual embedding E_v . Since the Transformer is insensitive to position, but the human pose estimation task is sensitive to position, we incorporate position embeddings E_{pos} into visual embeddings E_v to obtain visual tokens X_v as follows:

$$X_v = E_v + E_{pos}, \text{ where } X_v, E_v, E_{pos} \in \mathbb{R}^{N_{vis} \times d}, \quad (1)$$

where d is the dimension of hidden embeddings.

3.2 Coarse Encoder

The role of the coarse encoder is to extract shallow features for sparse keypoint tokens and part tokens to facilitate later grouping, pruning, and the introduction of dense keypoint tokens. It consists of Transformer layers, as shown in Fig. 3.

Transformer Layer. The Transformer layer [30] comprises two sub-layers: Multi-Head Self-Attention (MHSA) and Feed Forward Network (FFN). MHSA is a self-attention mechanism designed to capture dependencies between sequence elements. By attending to different aspects of sequence, MHSA effectively captures complex patterns and relationships between tokens. Specifically, for a sequence \mathbf{X} , it generates query \mathbf{Q} , key \mathbf{K} , and value \mathbf{V} through three linear projections. Subsequently, we can perform attention operation:

$$\text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i) = \mathbf{A}_i \mathbf{V}_i \quad (2)$$

$$= \text{Softmax} \left(\frac{\mathbf{Q}_i \mathbf{K}_i^T}{\sqrt{\hat{d}}} \right) \mathbf{V}_i, \quad (3)$$

where \mathbf{A}_i denotes the attention map of the i^{th} head. The Multi-Head Attention performs attention operations simultaneously in h different subspaces. The results are then concatenated and fed into a linear mapping to obtain outputs.

Human-2-Keypoint. In the shallow layers of TokenPose, each keypoint token focuses on the estimated human body, not the respective keypoints. To minimize computation, we only use a human token $\mathbf{X}_h \in \mathbb{R}^{1 \times d}$ to extract the human features in the shallow layers. The input to the coarse encoder is $\mathbf{X}^0 = [\mathbf{X}_h, \mathbf{X}_v] \in \mathbb{R}^{(N_{vis}+1) \times d}$, where $[\dots]$ denotes the concatenate operation. In the middle of the coarse encoder, we convert the human token \mathbf{X}_h into sparse keypoint tokens \mathbf{X}_s and part tokens \mathbf{X}_p through the Human-2-Keypoint (H2K):

$$\mathbf{X}_s = \mathbf{X}_h + \mathbf{E}_s, \mathbf{X}_p = \mathbf{X}_h + \mathbf{E}_p, \quad (4)$$

where $\mathbf{E}_s, \mathbf{X}_s \in \mathbb{R}^{N_s \times d}$ denotes sparse keypoints' learnable embeddings and tokens, $\mathbf{E}_p, \mathbf{X}_p \in \mathbb{R}^{5 \times d}$ denotes parts' learnable embeddings and tokens (for face, both hands, both feet) and N_s denotes the number of sparse keypoints, which is the total number of body keypoints. Additionally, we employ a pruning technique to remove background visual tokens based on the attention maps of the human token to improve efficiency. Further details can be found in Sec. 3.5.

3.3 Coarse-to-Fine Transition

The Coarse-to-Fine Module plays a role in introducing dense keypoints, grouping all keypoints, and selecting appropriate masked visual tokens for each group.

Group Operation. We group tokens to prune visual tokens more efficiently. In whole-body pose estimation, we incorporate the learnable dense keypoint embeddings \mathbf{E}_d with corresponding part tokens to obtain dense keypoint tokens \mathbf{X}_d . Next, we divide all keypoints into three groups: head, upper body, and lower body, based on the location of each keypoint. Considering the definition of the whole-body pose estimation task, it is evident that the face has the most keypoints, but the smallest area. On the contrary, the lower body has the fewest keypoints, but the largest area. Consequently, after grouping, we can prioritize pruning more visual tokens for the face and retaining more for the lower body

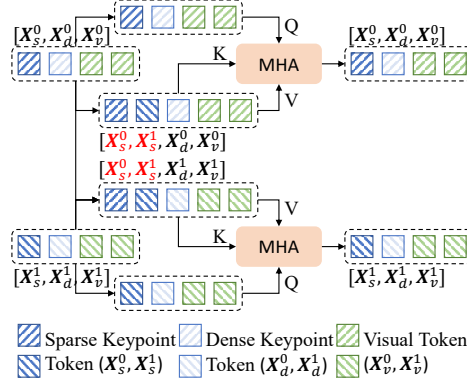


Fig. 4: An example of the interaction between two groups via MHGA, where different patterns indicate different groups and red represents sharing.

to ensure balanced computational efficiency for each group and enhance pruning efficiency without significantly compromising the performance.

Mask Operation. To ensure that visual tokens can adapt to the modeling requirements of each group, we enhance them through channel attention [11]. Specifically, we employ an MLP to derive masks for each visual token as follows:

$$\mathbf{M}^0, \mathbf{M}^1, \mathbf{M}^2 = \text{Sigmoid}(\text{MLP}(\mathbf{X}_v)), \quad (5)$$

where $\mathbf{M}^0, \mathbf{M}^1, \mathbf{M}^2 \in \mathbb{R}^{N_v \times d}$ denotes the visual token masks corresponding to three groups, and N_v denotes the number of remaining visual tokens. Afterward, we can obtain visual tokens for each group as follows:

$$\mathbf{X}_v^j = \mathbf{X}_v \cdot \mathbf{M}^j, \quad (6)$$

where j denotes the j^{th} group, \cdot denotes element-wise multiplication.

3.4 Fine Encoder

The Fine Encoder consists of Group Transformer layers. We introduce a pruning layer in the middle of the Fine Encoder, reducing visual tokens in each group to enhance efficiency. Additionally, following SimCC [14], we treat the keypoint localization problem as a classification task by discretizing the continuous horizontal and vertical axes into multiple bins. Consequently, we feed all keypoint tokens into the horizontal and vertical classifiers to predict 1D heatmaps.

Group Transformer Layer. In the Group Transformer Layer, we introduce Multi-Head Group-Attention (MHGA) instead of Multi-Head Self-Attention. MHGA aims to enhance computational efficiency by parallel computing attention only between the intra-group tokens. However, to enable interaction with inter-group information, we incorporate shared sparse keypoint tokens \mathbf{X}_s as

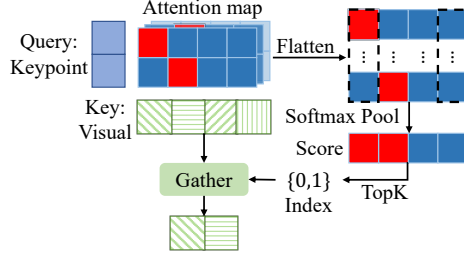


Fig. 5: Overview of the pruning process, where different patterns represent different visual tokens. The red indicates high scores, and the blue indicates low scores.

global information during the key and value calculation within each group. The specific process of MHGA is shown in Fig. 4. For instance, when computing the MHGA of the j^{th} group in the i^{th} head, three linear projections are still employed to obtain the query \mathbf{Q}_i^j , key \mathbf{K}_i^j , and value \mathbf{V}_i^j :

$$\mathbf{Q}_i^j = \mathbf{X}^j \mathbf{W}_i^Q, \quad (7)$$

$$\mathbf{K}_i^j = [\mathbf{X}_s, \mathbf{X}_d^j, \mathbf{X}_v^j] \mathbf{W}_i^K, \quad (8)$$

$$\mathbf{V}_i^j = [\mathbf{X}_s, \mathbf{X}_d^j, \mathbf{X}_v^j] \mathbf{W}_i^V. \quad (9)$$

The attention operation is then performed similarly as for the MHSA, utilizing Eq. (2). The final output of the MHGA is obtained by concatenating and projecting the outcomes from the multiple heads.

3.5 Global Perceived Pruning

Pruning. When visual tokens are input to estimate a person’s pose, the attention is solely on that individual. Consequently, the visual tokens representing the background or other persons are redundant. Therefore, it is crucial to prune redundant visual tokens to enhance efficiency. Inspired by PPT [19], we utilize attention maps as the foundation for pruning. As shown in Fig. 5, attention maps employ the remaining visual tokens as keys during each pruning stage. However, each group takes their keypoint tokens as queries. Multiple attention maps exist due to the multi-head attention or multiple keypoint tokens. Therefore, pooling is necessary to calculate the importance score for each visual token. Since different heads compute attention scores in different subspaces, some significant visual tokens may only receive high attention scores in a few. Therefore, we propose to calculate the score through softmax pooling as follows:

$$\mathbf{S} = \sum_i^h \sum_k^{N_k} \mathbf{W}_i \mathbf{A}_i^k, \text{ where } \mathbf{W}_i = \frac{e^{\mathbf{A}_i^k}}{\sum_i^h \sum_k^{N_k} e^{\mathbf{A}_i^k}}, \quad (10)$$

$\mathbf{A}_i^k \in \mathbb{R}^{N_v}$ denotes the attention weight of the k^{th} keypoint in the i^{th} head, $\mathbf{S} \in \mathbb{R}^{N_v}$ denotes the importance scores, h denotes the heads number, N_v denotes the

number of remaining visual tokens, and N_k denotes the number of keypoints in the group. We can then select the top \hat{N}_v visual tokens and prune the remaining ones. The number of retained visual tokens can be calculated as follows:

$$\hat{N}_v = (1 - \alpha)N_{vis} - N_d, \quad (11)$$

where α denotes the pruning rate, N_{vis} denotes the number of original visual tokens, and N_d denotes the number of dense keypoint tokens within the group. Such a pruning technique ensures that the number of tokens in each group is approximately equal, thereby enhancing efficiency.

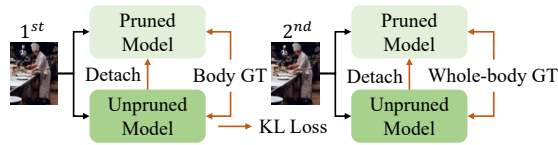


Fig. 6: Overview of the training strategy. The left part indicates the first stage for HPE, and the right part indicates the second stage for whole-body pose estimation.

Global Perceived Loss. Since the pruned model can only observe local information, it is easy to fall into local optimality. Therefore, we use pruned loss L_{pruned} to supervise the pruned model while using unpruned loss $L_{unpruned}$ to supervise the unpruned model with shared parameters. Because the unpruned model can observe the global information, we transfer the knowledge from the unpruned model to the pruned model through the Global-2-Local loss L_{G2L} , enhancing the pruned model’s global awareness. Thus, we train the model with the Global Perceived loss L_{GP} as follows:

$$L_{GP} = L_{pruned} + L_{unpruned} + L_{G2L}. \quad (12)$$

Curriculum Learning. Since GTPT is a new architecture designed for efficient human pose estimation, it has not been pre-trained on large-scale datasets. Directly estimating the whole-body pose is a significant challenge due to the numerous keypoints involved. Therefore, we employ curriculum learning, enabling the model to gradually learn how to locate all keypoints of the whole body in an easy-to-difficult manner. As shown in Fig. 6, we begin by training the model on simple body pose estimation. Then, we introduce part tokens and dense keypoint tokens to learn whole-body pose estimation. This approach effectively reduces the learning difficulty and enhances performance.

4 Experiments

4.1 Settings

Dataset & Evaluation Metrics. COCO [17] and COCO-WholeBody [13] are the most popular for human and whole-body pose estimation, respectively. Both

Method	GFLOPs	Params	AP	AR
CNN-based Methods				
SimBa.-Res50 [37]	8.9	34M	70.4	76.3
SimBa.-Res101 [37]	12.4	53M	71.4	77.1
SimBa.-Res152 [37]	15.7	68.6M	72.0	77.8
HRNet-W32 [27]	7.1	28.5M	74.4	79.8
HRNet-W48 [27]	14.6	63.6M	75.1	80.4
Lite-HRNet-18 [45]	0.2	1.1M	64.8	71.2
Lite-HRNet-30 [45]	0.3	1.8M	67.2	73.3
EfficientPose-B [1]	1.1	3.3M	71.1	-
EfficientPose-C [1]	1.6	5.0M	71.3	-
Transformer-based Methods				
TransPose-R-A4 [42]	8.9	6.0M	72.6	78.0
TransPose-H-S [42]	10.2	8.0M	74.2	79.5
TokenPose-S-v1 [15]	2.4	6.6M	72.5	78.0
TokenPose-B [15]	6.0	13.5M	74.7	80.0
DistilPose-S [44]	2.4	5.4M	71.6	-
DistilPose-L [44]	10.3	21.3M	74.4	-
PPT-S [19]	2.0	6.6M	72.2	77.8
PPT-B [19]	5.6	13.5M	74.4	79.6
GTPT-T	0.7	2.4M	71.1	76.6
GTPT-S	1.6	5.4M	73.6	78.9
GTPT-B	3.6	8.3M	74.9	80.0

Table 1: Comparison on COCO val. The input size is 256×192 .

Method	Input Size	GFLOPs	Whole-body		Body	Foot	Face	Hand
			AP	AR	AP	AR	AP	AR
SN [9]	N/A	272.3	32.7	45.6	42.7 58.3	9.9 36.9	64.9 69.7	40.8 58.0
OpenPose [2]	N/A	451.1	44.2	52.3	56.3 61.2	53.2 64.5	76.5 84.0	38.6 43.3
PAF [3]	512×512	329.1	29.5	40.5	38.1 52.6	5.3 27.8	65.6 70.1	35.9 52.8
AE [24]	512×512	212.4	44.0	54.5	58.0 66.1	57.7 72.5	58.8 65.4	48.1 57.4
DeepPose [29]	384×288	17.3	33.5	48.4	44.4 56.8	36.8 53.7	49.3 66.3	23.5 41.0
SimBa. [37]	384×288	20.4	57.3	67.1	66.6 74.7	63.5 76.3	73.2 81.2	53.7 64.7
HRNet [27]	384×288	16.0	58.6	67.4	70.1 77.3	58.6 69.2	72.7 78.3	51.6 60.4
PVT [34]	384×288	19.7	58.9	68.9	67.3 76.1	66.0 79.4	74.5 82.2	54.5 65.4
FastPose50-dcn-si [7]	256×192	6.1	59.2	66.5	70.6 75.6	70.2 77.5	77.5 82.5	45.7 53.9
ZoomNet [13]	384×288	28.5	63.0	74.2	74.5 81.0	60.9 70.8	88.0 92.4	57.9 73.4
ZoomNAS [38]	384×288	18.0	65.4	74.4	74.0 80.7	61.7 71.8	88.9 93.0	62.5 74.0
ViTPose+-S [40]	256×192	5.4	54.4	-	71.6 -	72.1 -	55.9 -	45.3 -
ViTPose+-H [40]	256×192	122.9	61.2	-	75.9 -	77.9 -	63.3 -	54.7 -
RTMPose-m [12]	256×192	2.2	58.2	67.4	67.3 75.0	61.5 75.2	81.3 87.1	47.5 58.9
RTMPose-l [12]	256×192	4.5	61.1	70.0	69.5 76.9	65.8 78.5	83.3 88.7	51.9 62.8
GTPT-T	256×192	0.8	54.9	65.6	67.6 75.9	64.9 77.4	75.4 84.1	38.3 49.9
GTPT-S	256×192	2.0	59.6	69.9	71.0 78.7	70.4 82.2	81.0 87.6	45.4 57.0
GTPT-B	256×192	4.0	61.7	71.4	72.0 79.5	73.0 84.0	84.2 89.6	47.9 59.3

Table 2: Comparison on COCO-WholeBody validation set.

datasets contain the same images, but the labeling differs. They both contain more than 200,000 images and 250,000 human instances. Each person in COCO only labeled 17 keypoints, but COCO-WholeBody labeled more fine-grained keypoints on the face, hands, and feet, so labeled 133 keypoints for each person. We follow the standard splitting of `train2017` and `val2017`. There are $118k$ images in the train set and $5k$ in the validation set. Meanwhile, we utilize a two-stage

top-down approach as the pipeline, which involves detecting the human by a detector first and then cropping it out to estimate its keypoints. In accordance with [37], we employ a person detector with an accuracy of 56.4% for the validation set. The standard evaluation metric for the COCO is the average precision (AP), which is calculated based on Object Keypoint Similarity (OKS).

Implementation Details. For both training stages, we adopt the same setting. Specifically, we employ 4 GPUs to conduct experiments based on pytorch. Random initialization without pre-training necessitates extended training epochs for fitting GTPT better. We use Adam as the optimizer, which has a learning rate of 1×10^{-3} decreased to 1×10^{-4} at the 200th epoch, to 1×10^{-5} at the 260th epoch and ended at the 300th; β_1 and β_2 are 0.9 and 0.999; weight decay is 10^{-4} .

4.2 Results

To showcase the scalability of GTPT, we developed three model variants, ranging from the smallest to the largest: GTPT-T, GTPT-S, and GTPT-B.

COCO. The results, presented in Table Tab. 1, underscore the efficiency of our method. Traditional convolutional methods demand numerous parameters, high computational costs, and pre-training on large datasets for better performance. In contrast, GTPT-T and GTPT-B, despite similar performance to SimBa-Rse101 and HRNet-W48, require just 5% and 25% FLOPs, respectively, even with random initialization. Although the efficiently designed Lite-HRNet and EfficientPose drastically reduce the number of parameters and computation. There is a significant loss in performance, and our method can achieve better results with a similar amount of computation. For example, the FLOPs of GTPT-S and EfficientPose-C are roughly the same, but we outperform them by 2.3 AP. For the Transformer-based approach, it is possible to maintain a small number of parameters and still perform better in general. Therefore, our approach inherits this advantage and can get better results with fewer FLOPs. For example, when comparing GTPT-S with the pruned PPT-S and the unpruned TokenPose-S, our FLOPs are lower, but the performance is 1.1 and 1.4 AP higher.

COCO-WholeBody. The results are shown in Tab. 2. In the table, most methods are pre-trained on large-scale datasets (such as ZoomNAS and RTMPose), unlike GTPT, which uses random initialization. As a Transformer-based approach, GTPT’s complexity scales quadratically with input length. Therefore, we adopt 256×192 in GTPT for efficient whole-body estimation. Since group-based pruning, we can achieve better results with less computational overhead. Compared to most methods that require high-resolution inputs, our approach achieves better results with 256×192 . GTPT-T achieves better results with only 15% FLOPs to ViTPose+-S. Our method still has a comparative advantage over equally efficiently designed methods. For example, GTPT-S performs 1.4 AP higher with lower FLOPs to RTMPose-m.

	Group	Mask	GFLOPs	Body	Foot	Face	Hand	Whole
①	✗	✗	1.77	69.2	68.0	79.8	44.3	58.2
②	✓	✗	1.98	70.6	69.7	80.6	44.7	59.1(+0.9)
③	✓	✓	1.99	71.0	70.4	81.0	45.4	59.6(+1.4)

Table 3: Ablation study of grouping and masking.

Method	GFLOPs	Body	Foot	Face	Hand	Whole
w/o Grouping	1.77	69.2	68.0	79.8	44.3	58.2
MHSA	1.99	70.2	69.4	80.4	44.0	58.8(+0.6)
MHGA	1.99	71.0	70.4	81.0	45.4	59.6(+1.4)

Table 4: Ablation study of MHGA.

4.3 Ablation Study

To exemplify the capability of our method in localizing a large number of keypoints, we uniformly adopt COCO-Wholebody as our dataset and employ group-based and global perceived pruning by default for ablation experiments.

Group & Mask. The purpose of grouping and masking is to enhance the model’s performance by extracting specific features for different groups. To validate their effectiveness, we conducted several ablation experiments. From the results in Tab. 3, we can conclude that excluding grouping (① *vs.* ②) and masking (② *vs.* ③) may reduce computation while leading to a drop in performance. However, when keypoints are grouped (① *vs.* ②), the corresponding 0.9 AP performance improvement justifies the 0.21 increase in GFLOPs. Additionally, masking the visual tokens of different groups (② *vs.* ③) results in a negligible increase in parameters and computation, yet it yields a remarkable 1.4 AP performance improvement compared to the baseline.

MHGA. When we group keypoint tokens and adopt MHSA, the interaction between keypoints from different groups is limited, affecting the modeling of overall keypoint relationships. Therefore, we propose MHGA, which aims to interact with sparse keypoint tokens as global information while maintaining efficiency. To verify the effectiveness of MHGA, we conducted two experiments. The experimental results in Tab. 4 reveal that solely performing grouping with MHSA can only bring 0.6 AP improvement. However, MHGA improves by 1.4 AP compared to no grouping, highlighting the importance of information interaction among different groups. Meanwhile, the difference in computation efficiency between MHSA and MHGA is negligible.

Keypoint Token Introduction Method. To improve efficiency while maintaining performance, we propose a coarse-to-fine approach to introducing keypoint tokens, starting from 1 human token and gradually increasing to a small number of sparse keypoint tokens and part tokens, and then to a large number of dense keypoint tokens. To validate the effectiveness, we conducted three experiments. Upon reviewing Tab. 5, it is evident that the choice of the keypoint introduction approach has a negligible impact on the number of parameters. Directly introducing all keypoints yields the highest performance but incurs a

Method	Params	GFLOPs	Body	Foot	Face	Hand	Whole
Dense	5.42M	2.26	71.0	70.7	80.8	45.7	59.8
Sparse-Dense	5.42M	2.02(-11%)	70.8	70.7	80.7	45.0	59.5
Human-Sparse-Dense	5.42M	1.99(-12%)	71.0	70.4	81.0	45.4	59.6

Table 5: Ablation study of keypoint token introduction method.

	Pruning	G&M	GFLOPs	Body	Foot	Face	Hand	Whole
① Direct	✗		1.77	68.0	66.5	79.0	42.7	56.6
② GP Loss	✗		1.77	69.2	68.0	79.8	44.3	58.2
③ Without	✓		3.53	70.9	70.7	80.8	44.9	59.4
④ Direct	✓	1.99(-44%)	70.5	70.3	80.4	44.4	59.2	
⑤ GP Loss	✓	1.99(-44%)	71.0	70.4	81.0	45.4	59.6	

Table 6: Ablation study of group-based and global perceived pruning, where G&M denotes grouping and masking.

higher computational overhead due to the excessive number of keypoints. In contrast, the Sparse-Dense approach effectively reduces computation with a slight impact on performance. Moreover, the Human-Sparse-Dense approach further decreases computation while rebounding in performance.

Group-based and Global Perceived Pruning. The pruned model is prone to fall into local optimality because it can only observe local information. However, the global awareness capability is essential for the pose estimation task, so we propose the global perceived loss. To verify the effectiveness of our proposed global perceived loss and whether grouping helps pruning, we conducted several experiments. Tab. 6 reveals that pruning is crucial for efficiency (③ *vs.* ⑤), which directly reduces 44% computation overhead. Notably, direct pruning with the proposed G&M does not cause much performance drop (③ *vs.* ④). In contrast, direct pruning without G&M results in significant performance loss (① *vs.* ②), suggesting that our model exhibits a high tolerance for pruning. When applying our proposed global perceived loss for pruning optimization, the model outperforms the unpruned model across most metrics (③ *vs.* ⑤), thus substantiating the effectiveness of global perceived loss.

5 Conclusion

This work proposes the Group-based Token Pruning Transformer (GTPT) for efficient human pose estimation, especially whole-body pose estimation with numerous keypoints. GTPT reduces redundancy by grouping keypoint tokens and pruning visual tokens, improving performance while decreasing computation. To interact with inter-group information, GTPT performs global interaction using MHGA with less computational effort. To mitigate the impact of numerous keypoints, GTPT introduces keypoints in a coarse-to-fine manner, further reducing computational overhead. We conduct extensive experiments on COCO and COCO-WholeBody to demonstrate effectiveness.

References

1. Bukschat, Y., Vetter, M.: Efficientpose: An efficient, accurate and scalable end-to-end 6d multi object pose estimation approach. arXiv preprint arXiv:2011.04307 (2020)
2. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis & Machine Intelligence* **43**(01), 172–186 (2021)
3. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7291–7299 (2017)
4. Chen, H., Feng, R., Wu, S., Xu, H., Zhou, F., Liu, Z.: 2d human pose estimation: A survey. arXiv preprint arXiv:2204.07370 (2022)
5. Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems* **34**, 17864–17875 (2021)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
7. Fang, H.S., Li, J., Tang, H., Xu, C., Zhu, H., Xiu, Y., Li, Y.L., Lu, C.: Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022)
8. Garau, N., Bisagno, N., Bródka, P., Conci, N.: Deca: Deep viewpoint-equivariant human pose estimation using capsule autoencoders. arXiv preprint arXiv:2108.08557 (2021)
9. Hidalgo, G., Raaj, Y., Idrees, H., Xiang, D., Joo, H., Simon, T., Sheikh, Y.: Single-network whole-body pose estimation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6982–6991 (2019)
10. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
11. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
12. Jiang, T., Lu, P., Zhang, L., Ma, N., Han, R., Lyu, C., Li, Y., Chen, K.: Rtm-pose: Real-time multi-person pose estimation based on mmpose. arXiv preprint arXiv:2303.07399 (2023)
13. Jin, S., Xu, L., Xu, J., Wang, C., Liu, W., Qian, C., Ouyang, W., Luo, P.: Whole-body human pose estimation in the wild. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16. pp. 196–214. Springer (2020)
14. Li, Y., Yang, S., Liu, P., Zhang, S., Wang, Y., Wang, Z., Yang, W., Xia, S.T.: Simcc: A simple coordinate classification perspective for human pose estimation. In: European Conference on Computer Vision. pp. 89–106. Springer (2022)
15. Li, Y., Zhang, S., Wang, Z., Yang, S., Yang, W., Xia, S.T., Zhou, E.: Tokenpose: Learning keypoint tokens for human pose estimation. arXiv preprint arXiv:2104.03516 (2021)
16. Li, Z., Ye, J., Song, M., Huang, Y., Pan, Z.: Online knowledge distillation for efficient pose estimation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11740–11750 (2021)

17. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
18. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
19. Ma, H., Wang, Z., Chen, Y., Kong, D., Chen, L., Liu, X., Yan, X., Tang, H., Xie, X.: Ppt: token-pruned pose transformer for monocular and multi-view human pose estimation. In: European Conference on Computer Vision. pp. 424–442. Springer (2022)
20. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV). pp. 116–131 (2018)
21. Mao, W., Ge, Y., Shen, C., Tian, Z., Wang, X., Wang, Z.: Tfpote: Direct human pose estimation with transformers. arXiv preprint arXiv:2103.15320 (2021)
22. Mao, W., Ge, Y., Shen, C., Tian, Z., Wang, X., Wang, Z., den Hengel, A.v.: Poseur: Direct human pose regression with transformers. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI. pp. 72–88. Springer (2022)
23. Neff, C., Sheth, A., Furgurson, S., Tabkhi, H.: Efficienthrnet: Efficient scaling for lightweight high-resolution multi-person pose estimation. arXiv preprint arXiv:2007.08090 (2020)
24. Newell, A., Huang, Z., Deng, J.: Associative embedding: End-to-end learning for joint detection and grouping. In: Advances in Neural Information Processing Systems (2017)
25. Osokin, D.: Real-time 2d multi-person pose estimation on cpu: Lightweight openpose. arXiv preprint arXiv:1811.12004 (2018)
26. Shen, X., Yuan, G., Niu, W., Ma, X., Guan, J., Li, Z., Ren, B., Wang, Y.: Towards fast and accurate multi-person pose estimation on mobile devices. arXiv preprint arXiv:2106.15304 (2021)
27. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5693–5703 (2019)
28. Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. *Advances in neural information processing systems* **27** (2014)
29. Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1653–1660 (2014)
30. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017)
31. Vats, A., Anastasiu, D.C.: Key point-based driver activity recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3274–3281 (2022)
32. Wang, H., Liu, J., Tang, J., Wu, G.: Lightweight super-resolution head for human pose estimation. arXiv preprint arXiv:2307.16765 (2023)
33. Wang, M., Tighe, J., Modolo, D.: Combining detection and tracking for human pose estimation in videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11088–11096 (2020)

34. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 568–578 (2021)
35. Wang, Y., Li, M., Cai, H., Chen, W.M., Han, S.: Lite pose: Efficient architecture design for 2d human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13126–13136 (2022)
36. Wehrbein, T., Rudolph, M., Rosenhahn, B., Wandt, B.: Probabilistic monocular 3d human pose estimation with normalizing flows. arXiv preprint arXiv:2107.13788 (2021)
37. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Proceedings of the European conference on computer vision (ECCV). pp. 466–481 (2018)
38. Xu, L., Jin, S., Liu, W., Qian, C., Ouyang, W., Luo, P., Wang, X.: Zoomnas: searching for whole-body human pose estimation in the wild. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(4), 5296–5313 (2022)
39. Xu, Y., Zhang, J., Zhang, Q., Tao, D.: Vitpose: Simple vision transformer baselines for human pose estimation. arXiv preprint arXiv:2204.12484 (2022)
40. Xu, Y., Zhang, J., Zhang, Q., Tao, D.: Vitpose+: Vision transformer foundation model for generic body pose estimation. arXiv preprint arXiv:2212.04246 (2022)
41. Yadav, S.K., Luthra, A., Tiwari, K., Pandey, H.M., Akbar, S.A.: Arfdnet: An efficient activity recognition & fall detection system using latent feature pooling. Knowledge-Based Systems **239**, 107948 (2022)
42. Yang, S., Quan, Z., Nie, M., Yang, W.: Transpose: Keypoint localization via transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11802–11812 (2021)
43. Yang, Z., Zeng, A., Yuan, C., Li, Y.: Effective whole-body pose estimation with two-stages distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4210–4220 (2023)
44. Ye, S., Zhang, Y., Hu, J., Cao, L., Zhang, S., Shen, L., Wang, J., Ding, S., Ji, R.: Distilpose: Tokenized pose regression with heatmap distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2163–2172 (2023)
45. Yu, C., Xiao, B., Gao, C., Yuan, L., Zhang, L., Sang, N., Wang, J.: Lite-hrnet: A lightweight high-resolution network. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10440–10450 (2021)
46. Yuan, Y., Fu, R., Huang, L., Lin, W., Zhang, C., Chen, X., Wang, J.: Hrformer: High-resolution vision transformer for dense predict. Advances in Neural Information Processing Systems **34**, 7281–7293 (2021)
47. Zeng, A., Sun, X., Yang, L., Zhao, N., Liu, M., Xu, Q.: Learning skeletal graph neural networks for hard 3d pose estimation. arXiv preprint arXiv:2108.07181 (2021)
48. Zeng, W., Jin, S., Liu, W., Qian, C., Luo, P., Ouyang, W., Wang, X.: Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11101–11111 (2022)
49. Zhang, F., Zhu, X., Ye, M.: Fast human pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3517–3526 (2019)
50. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)

51. Zou, S., Guo, C., Zuo, X., Wang, S., Wang, P., Hu, X., Chen, S., Gong, M., Cheng, L.: Eventhpe: Event-based 3d human pose and shape estimation. arXiv preprint arXiv:2108.06819 (2021)