SHaRPose: Sparse High-Resolution Representation for Human Pose Estimation

Xiaoqi An[1,2], Lin Zhao[1,2*], Chen Gong[1], Nannan Wang[2], Di Wang[2], Jian Yang[1*]

[1]PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education
Jiangsu Key Lab of Image and Video Understanding for Social Security
School of Computer Science and Engineering, Nanjing University of Science and Technology
[2] State Key Laboratory of Integrated Services Networks, Xidian University
{xiaoqi.an, linzhao, chen.gong, csjyang}@njust.edu.cn, {nnwang, wangdi}@xidian.edu.cn

Abstract

High-resolution representation is essential for achieving good performance in human pose estimation models. To obtain such features, existing works utilize high-resolution input images or fine-grained image tokens. However, this dense high-resolution representation brings a significant computational burden. In this paper, we address the following question: "Only sparse human keypoint locations are detected for human pose estimation, is it really necessary to describe the whole image in a dense, high-resolution manner?" Based on dynamic transformer models, we propose a framework that only uses Sparse High-resolution Representations for human Pose estimation (SHaRPose). In detail, SHaRPose consists of two stages. At the coarse stage, the relations between image regions and keypoints are dynamically mined while a coarse estimation is generated. Then, a quality predictor is applied to decide whether the coarse estimation results should be refined. At the fine stage, SHaRPose builds sparse high-resolution representations only on the regions related to the keypoints and provides refined high-precision human pose estimations. Extensive experiments demonstrate the outstanding performance of the proposed method. Specifically, compared to the state-of-the-art method ViTPose, our model SHaRPose-Base achieves 77.4 AP (+0.5 AP) on the *COCO* validation set and 76.7 AP (+0.5 AP) on the *COCO* test-dev set, and infers at a speed of 1.4× faster than ViTPose-Base. Code is available at https://github.com/AnxQ/sharpose.

Introduction

2D human pose estimation (HPE) is a fundamental task in the field of computer vision. Its main goal is to locate a set of anatomical keypoints that correspond to the human body's joints and limbs in an image. HPE has been well studied (Guo 2020; Zhang et al. 2021; Chang et al. 2020) and forms the foundation for many downstream tasks such as action recognition (Kawai, Yoshida, and Liu 2022; Chao et al. 2017; Xu et al. 2022a; Duan et al. 2022) and abnormal behavior detection (Tang et al. 2021; Qiu et al. 2022). Due to its potential applications in the real world, HPE remains an active area of research (Niemirepo, Viitanen, and Vanne 2020; Yu et al. 2021; Zhang, Zhu, and Ye 2019; Li et al. 2022, 2021c; Jiang et al. 2023).

*Corresponding authors.

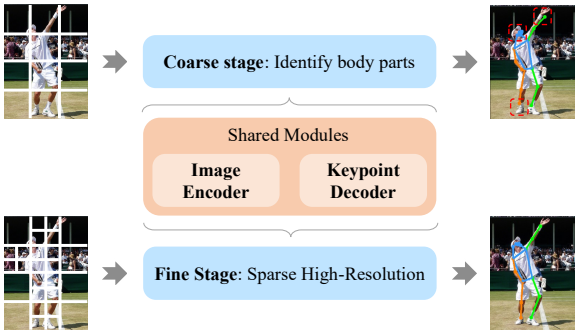

Figure 1: A brief view of SHaRPose. The coarse stage selects image parts contributed to the keypoints, and the fine stage builds high-resolution representations upon them.

In recent years, great progress has been made in human pose estimation (Toshev and Szegedy 2014; Newell, Yang, and Deng 2016; Xiao, Wu, and Wei 2018; Wang et al. 2021a; Chen et al. 2017). Most of the leading methods output heatmaps and then take the peak of heatmaps as the keypoint position. Hence, similar to other dense prediction tasks such as semantic segmentation (Ke et al. 2022; Guo et al. 2022) and depth estimation (Shen et al. 2021; Luo et al. 2020), it's necessary to obtain high-resolution representation to ensure the inference accuracy (Badrinarayanan, Kendall, and Cipolla 2017; Lin et al. 2017; Chen et al. 2018). For example, Stacked Hourglass (Newell, Yang, and Deng 2016) achieves high-quality image representation by stacking a symmetric encoding-decoding structure, while HRNet (Wang et al. 2021a) utilizes multiple parallel convolution branches to preserve high-resolution feature representations. ViTPose (Xu et al. 2022b) achieves notable performance using an 8×8 fine-grained patch splitting setting.

However, it is observed that increasing the resolution of feature representation (*i.e.*, the number of image tokens for transformer-based methods) results in an intensive computational burden. As shown in Table.1, this is particularly significant in Transformer-based methods because the complexity of Transformers is quadratic to the number of tokens (Khan et al. 2022). In this paper, we aim to improve the efficiency of transformer-based models for human pose estimation, and we think about the following question: *Since we*

Model	Input size	FPS	FLOPS	AP
HRNet	256×192	194	15.8	75.1
	384×288	152(-21%)	35.5(+125%)	76.3
ViTPose	256×192	340	18.6	75.8
	384×288	143(-58%)	44.1(+136%)	76.9

Table 1: Computational cost for high-resolution input

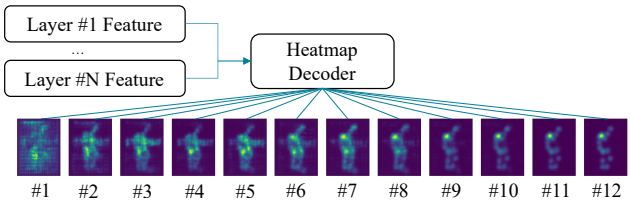

Figure 2: Decoder's response of ViTPose. Each heatmap is generated by feeding the output of each intermediate Transformer layer to the heatmap decoder.

only want the keypoint locations, which are sparse relative to the entire image, do we truly need high-resolution feature representation for all contents?

Based on this thinking, we conduct experiments using ViTPose (Xu et al. 2022b), as shown in Fig.2. Each heatmap is obtained by redirecting the intermediate layer's output to the decoder. These heatmaps provide an intuitive visualization of the image regions that the decoder is focusing on. We can observe that only in the first few layers, the output of the Transformer causes a global response on the decoder, while in the subsequent layers, the decoder's response is clearly concentrated on the sparse local areas containing keypoints. This means that during the inference, a large part of the image tokens like those only containing background information do not provide effective context information. Thus, Focusing solely on keypoint-related image regions may be sufficient to achieve accurate estimation results. And this can significantly reduce the computation costs.

Inspired by this idea, we propose a method that only needs Sparse High-resolution Representation to do human Pose estimation, named SHaRPose. The framework is based on pure transformers and makes use of the correlation mining capabilities of Transformer (Bach et al. 2015; Chefer, Gur, and Wolf 2021; Liang et al. 2022) to identify significant image regions for keypoint detection.

An overview of our framework is illustrated in Fig.1. The inference process is divided into two stages: The initial stage of our network processes coarse-grained patches as inputs, leading to diminished computational expenses owing to the decreased token count. Then a quality predictor module is applied to judge the roughly predicted pose. If the module yields a high confidence score, the network inference terminates. If not, the input image is split into finer-grained patches and fed into the fine stage to get refined results. To avoid computational burden on redundant patches, only the image patches with strong correlations to keypoints are split into finer-grained patches, while patches with weaker correlations are retained in the coarse-grained state. Hence, the proposed approach prevents heavy computational loads caused by processing unnecessary high-resolution image patches.

Overall, the main contributions of this paper are as follows:

- SHaRPose proposes to use sparse high-resolution representations, which is the first time that a dynamic optimization strategy has been introduced into the pose estimation task as far as our knowledge goes.
- SHaRPose greatly improves the efficiency of pure transformer models in the task of pose estimation. We reduce 25% of GFLOPs and achieve a $1.4\times$ higher FPS compared to ViTPose-Base.
- SHaRPose shows competitive performance with much higher throughput than the existing state-of-the-art models on the MS *COCO* dataset. We achieve 77.4 AP (+0.5 AP) on *COCO* validation set and 76.7 AP (+0.5 AP) on *COCO* test-dev set compared to ViTPose-Base.

Related Works

Vision Transformer for Pose Estimation

Vision Transformers (ViT) crop and map 2D images or image feature representations into token tensors to model long-range dependencies. With the overwhelming performance of Transformers in various computer vision tasks (Dosovitskiy et al. 2021; Liu et al. 2021; Xia et al. 2022; Liu et al. 2022; Carion et al. 2020; Wang et al. 2021b), some works have introduced Transformers to pose estimation, because the capability of ViT to capture long-range dependencies is of notable value in modeling the structure of the human body (Ramakrishna et al. 2014; Tompson et al. 2014; Wei et al. 2016). PRTR (Li et al. 2021a) proposes a cascaded transformer structure to achieve end-to-end keypoint coordinate regression. TransPose (Yang et al. 2021) utilizes a transformer encoder to process feature maps and to produce interpretable heatmap responses. HRFormer (Yuan et al. 2021) adopts the structure of HRNet (Wang et al. 2021a) and inserts attention blocks into branches to achieve larger receptive fields. On the other hand, TFPose (Mao et al. 2021) uses a set of keypoint queries to regress coordinates from transformers, while TokenPose (Li et al. 2021b) proposes token-based heatmap representations to model the body parts explicitly. ViTPose (Xu et al. 2022b) explores the feasibility of using a plain transformer as the backbone network for pose estimation and achieves excellent prediction accuracy with the help of masked image modeling (He et al. 2022) and multi-dataset training.

In general, compared with the pure CNN-based methods (Wang et al. 2021a; Newell, Yang, and Deng 2016; Toshev and Szegedy 2014), the transformer-based models are more likely to achieve good results with the help of global attention. However, this also leads to a larger computation cost. In this paper, a sparse high-resolution representation mechanism is explored, which saves considerable computation while retaining the global modeling advantages and high precision of the transformer-based methods.

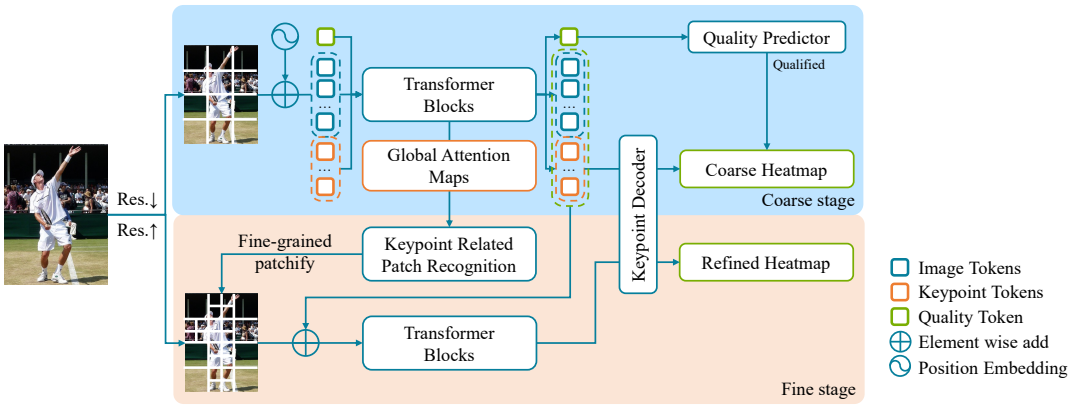

Figure 3: The overall structure of SHaRPose. The attention maps yielded by the transformer in the coarse stage is used for selecting keypoint-related patches in the fine stage. Only these keypoint-related patches are processed in finer granularity in the fine stage. The parameters of the Transformer blocks and the keypoint decoder are shared between the two stages.

Dynamic Vision Transformer

To mitigate the issue of computational resource consumption resulting from global feature interaction in Transformers, many methods have been proposed, among which dynamic optimization is one of the major categories.

The simplest approaches involve reducing the number of input tokens to Transformer by pruning them: DynamicViT (Rao et al. 2021) uses a lightweight detector to determine which tokens to keep, ToMe (Bolya et al. 2023) fuses similar tokens based on their similarity, and EviT (Liang et al. 2022) evaluates the importance of image blocks based on class attention. On the other hand, some methods gradually adjust the input granularity from a coarse level. QuadTree (Tang et al. 2022) obtains attention from different scales at each layer and performs cross-scale weighting to capture comprehensive representations, thus reducing the number of tokens involved in attention. DVT (Wang and Torresani 2022) uses adaptive patch size to reduce the calculation on easy samples. CF-ViT (Chen et al. 2023) designs two stages using different granularity patches and reorganizes the specific fine-grained tokens with the coarse-grained tokens to refine the prediction in the second stage.

The above-mentioned works have achieved good trade-offs between accuracy and performance. However, the success of these methods is mainly demonstrated in the classification task. In this work, we adapt dynamic transformers to the pose estimation task. Because retaining global context is helpful for human pose estimation, and discarding tokens may cause the model to produce biased predictions, we follow the second category of dynamic transformer methods, designing the framework in a coarse-to-fine manner.

Method

Overall Structure

As depicted in Fig.3, SHaRPose contains two stages with a shared keypoint decoder. The coarse stage consists of a Transformer and a quality predictor module. The fine stage includes a keypoint-related patch selection module and a Transformer sharing the same parameters as the one in the coarse stage.

In this section, we will present our framework stage-by-stage and give a detailed introduction to each module.

Coarse-Inference Stage

The goal of this stage is to capture relations between image regions and keypoints, as well as give a coarse inferred heatmap and decide whether the heatmap is accurate enough. To accomplish the objective, a set of keypoint tokens and a quality token are introduced as the queries.

Token Input Denote the input image $X \in \mathbb{R}^{H\times W\times C}$, given the specific patch size p_h, p_w and an input scaling factor s_c, we compose the input token sequence as follows:

$$\begin{aligned} X_c &= \text{Resample}(X) \in \mathbb{R}^{H\cdot s_c \times W\cdot s_c \times C} \\ X_0^c &= \left[v_0^1; v_0^2; \dots v_0^{N_c}; k_0^1; k_0^2; \dots k_0^M; q_0\right], \end{aligned} \tag{1}$$

where v_0^i is the visual token, obtained from the Re-sampled image X_c. First, X_c is split into $N_c = \frac{H\cdot s_c}{p_h} \times \frac{W\cdot s_c}{p_w}$ patches, then a linear projection $f : p \to v \in \mathbb{R}^D$ is applied to get the corresponding v_0^i. M is the number of keypoints, and $\{k_0^i \in \mathbb{R}^D\}_{i=1}^M$ are keypoint tokens from M learnable embeddings, representing the query of keypoints. $q_0 \in \mathbb{R}^D$ is a quality token also from a learnable embedding, which will be used to estimate the quality of the predicted human pose.

Transformer Encoder After the composition of the input token sequence, a K-layers transformer $\mathcal{V}$ (Dosovitskiy et al. 2021) is applied to obtain the output sequence:

$$X_K^c = \mathcal{V}(X_0^c) = \left[v_K^1; v_K^2; \dots v_K^{N_c}; k_K^1; k_K^2; \dots k_K^M; q_K\right]. \tag{2}$$

Keypoint Decoder The keypoint decoder builds heatmaps from M output keypoint tokens $\{k_K^i\}_{i=1}^M$ through an unified multiple linear projection module:

$$\mathbf{H}_i^c = \mathcal{D}(k_K^i) \in \mathbb{R}^{\hat{H}\times\hat{W}}, \tag{3}$$

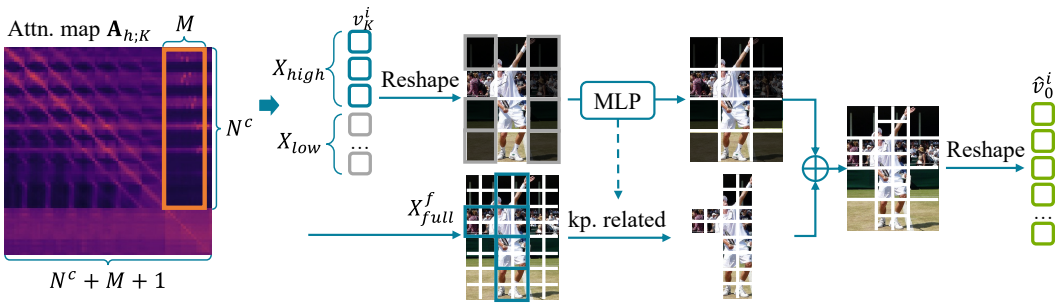

Figure 4: Compose the input of the fine stage. The attention scores $\hat{\mathbf{A}}_{h;k}$ between visual tokens and keypoint tokens are just part of the full attention matrix $\mathbf{A}_{h;k}$. Only high-score image patches (blue) are further split into fine-grained patches. An MLP is applied to incorporate the coarse-stage information into the fine stage.

where $\hat{H} \times \hat{W}$ is the heatmap size, $\hat{H}$ and $\hat{W}$ are both $1/4$ of the original image size H and W. Through the decoder, the coarse-predicted heatmaps $\{\mathbf{H}_i^c\}_{i=1}^M$ are acquired.

Quality Predictor Inspired by (Zhao et al. 2021; Fang et al. 2017), we use a learnable quality embedding q_0 to obtain the quality of the predicted keypoints by grubbing information from both visual tokens and keypoint tokens. Then, the quality predictor module produces the quality score of the prediction through the information fused in the quality token:

$$Q = \text{MLP}(q_K). \tag{4}$$

With the estimated quality score Q, we set a threshold Q_{thres}. Only if $Q < Q_{thres}$, the image will be split into finer-grained patches and processed in the fine stage. This allows the model to dynamically distinguish hard and easy samples. Therefore, the number of images that go through the fine stage can be reduced, which can further increase the throughput.

Fine-Inference Stage

In this stage, the model generates sparse high-resolution representations and makes high-precision predictions of poses by leveraging the attention obtained in the coarse stage.

Keypoint-Related Patch Recognition In order to decide which image regions need high-resolution feature representations, a kind of relevance score between image patches and keypoints is required. As shown in Fig.4, consider a slice of the attention matrix in a layer of the Transformer $\mathcal{V}$, which is defined as follows:

$$\hat{\mathbf{A}}_{h;k} = \left[\hat{\mathbf{a}}_{h;k}^{N^c+1}; \hat{\mathbf{a}}_{h;k}^{N^c+2}; \ldots; \hat{\mathbf{a}}_{h;k}^{N^c+M}\right] \in \mathbb{R}^{M \times N^c}, \tag{5}$$

where $\hat{\mathbf{a}}_{h;k}^{N^c+i}$ is the attention score vector between the keypoint token k_{k-1}^i and all coarse-grained image tokens $\{v_{k-1}^i\}_{i=1}^{N^c}$ at head h. This vector reflects the interaction between the keypoint token and the visual tokens. Following (Chen et al. 2023), we also use exponential moving average (EMA) to combine attentions from each Transformer layer:

$$\overline{\mathbf{A}}_{h;k} = \beta \cdot \overline{\mathbf{A}}_{h;k-1} + (1-\beta) \cdot \hat{\mathbf{A}}_{h;k}, \tag{6}$$

in which we set $\beta = 0.99$. Then we take the accumulated attention matrix of the last layer $\overline{\mathbf{A}}_{h;K}$ and mix the attention vector of each keypoint and different heads in the following form to get the final visual token correlation score:

$$\mathbf{s} = \frac{1}{HM} \sum_{h=1}^{H} \sum_{i=1}^{M} \overline{\mathbf{a}}_{h;K}^{N^c+i}, \tag{7}$$

where $\overline{\mathbf{a}}_{h;K}^{N^c+i}$ is the ith column of the matrix $\overline{\mathbf{A}}_{h;K}$, H and M denote the number of heads and the number of keypoints, respectively. According to $\mathbf{s}$, we can rank and select the image patches which are important to estimate human pose. As shown in Fig.4, We select a set X_{high} consisting of $N^h = \lfloor \alpha \cdot N^c \rfloor$ patches with higher scores from $\{v_K^i\}_{i=1}^{N^c}$, while the remaining patches form the set X_{low}.

Fine Inference To perform the fine stage inference, we first need to construct high-resolution representations. Similar to Eq.1 in Section , given the scaling ratio in the fine stage s_f, the full visual tokens can be obtained as $X_{full}^f = \{v_f^i\}_{i=0}^{N_f}$, where $N_f = \frac{H \cdot s_f}{p_h} \cdot \frac{W \cdot s_f}{p_w}$ is the number of all fine-grained image tokens.

The initial visual tokens of the fine stage $\{\hat{v}_0^i\}_{i=0}^{\hat{N}_f}$ can be constructed as follows: The first part is composed of tokens not closely associated with keypoints, which can be directly taken from X_{low}. The second part comprises tokens generated from X_{high}, which are more relevant to keypoints. Denote a singular visual token from X_{high} as v_K^j, which is further split into $N = (s_f/s_c)^2$ fine-grained tokens. The computation of the new input visual tokens is formulated as $\{\text{MLP}(v_K^j) + v_f^{c_i}\}_{i=0}^{N}$, where $\{v_f^{c_i}\}_{i=0}^{N}$ are fine image tokens from X_{full}^f at the corresponding location of v_K^j. Thus, the input token sequence can be formed by:

$$X_0^f = \left[\hat{v}_0^1; \hat{v}_0^2; \ldots \hat{v}_0^{\hat{N}_f}; k_0^1; k_0^2; \ldots k_0^M\right], \tag{8}$$

where $\hat{N}_f = N \cdot \lfloor \alpha \cdot N_c \rfloor + \lfloor (1-\alpha) \cdot N_c \rfloor$ is the number of visual tokens, k_0^i is the same initial keypoint token embedding as in Eq.1. We present the process of building the fine stage visual tokens in Fig.4.

Model	Input Size	Feat. Dim.	Depth	Patch Size	s_c	s_f	α	FPS	Params	GFLOPs
SHaRPose-Small	256×192	384	12	16×16	0.5	1.0	0.5	498.3	28.4M	4.9
SHaRPose-Small	384×288	384	12	16×16	0.5	1.0	0.5	395.3	48.3M	11.0
SHaRPose-Base	256×192	768	12	16×16	0.5	1.0	0.4	392.8	93.9M	17.1
SHaRPose-Base	384×288	768	12	16×16	0.5	1.0	0.3	196.6	118.1M	32.9

Table 2: Configurations of the instantiated SHaRPose models. We provide the detailed parameters for constructing both the Base and the Small models. And the specific model sizes are presented in the last columns of the table.

Then, similar to Eq.2, a transformer sharing the same parameters as the one in the coarse stage is applied to get the output of the fine stage by $X_K^f = \mathcal{V}(X_0^f)$. Finally, the keypoint tokens are fed into a shared decoder defined in Eq.3 to get the fine inferred heatmaps $\mathbf{H}_i^f = \mathcal{D}(k_K^i)$.

Loss Function

For training the network, we impose supervision both on the output heatmaps and the pose confidence that the quality predictor infers:

$$\mathcal{L} = \mathcal{L}_{heatmap} + \lambda \mathcal{L}_{qp}, \tag{9}$$

in which λ is a hyper-parameter to balance the loss terms. $\mathcal{L}_{heatmap}$ is the heatmap mean square error loss, including the coarse stage and the fine stage:

$$\mathcal{L}_{heatmap} = \frac{1}{M} \sum_{i}^{M} \left(\mathcal{L}_{mse}(\mathbf{H}_i^c, \mathbf{H}_i^{gt}) + \mathcal{L}_{mse}(\mathbf{H}_i^f, \mathbf{H}_i^{gt}) \right), \tag{10}$$

in which $\mathbf{H}_i^{gt}$ is the ground-truth heatmap. $\mathcal{L}_{qp}$ is an L2-norm loss between the quality predictor's output Q and the coarse stage's ground-truth OKS, which denotes the object keypoint similarity:

$$\mathcal{L}_{qp} = \left\| Q - \mathrm{OKS}^{gt} \right\|_2. \tag{11}$$

Experiments

Experiment Setup

Datasets We conduct experiments on *COCO* (Lin et al. 2014) and *MPII* (Andriluka et al. 2014) datasets. Following the customary strategy of Top-Down methods (Xiao, Wu, and Wei 2018; Wang et al. 2021a; Newell, Yang, and Deng 2016), we utilize the *COCO* 2017 dataset, which comprises 200k images and 250k person instances. The dataset is segregated into three subsets: train, valid, and test-dev, containing 150k, 5k, and 20k samples, respectively. We train our model on the train subset and test it on the valid and test-dev subsets. The *MPII* dataset, which comprises over 40k person instances and 25k images, is also employed for training and evaluation.

Evaluation Metrics Following (Wang et al. 2021a; Yuan et al. 2021; Xu et al. 2022b; Li et al. 2021b), we use the standard average precision (AP) as evaluation metric on the *COCO* dataset, which is calculated based on OKS. On the other hand, we perform head-normalized percentage of correct keypoint (PCKh) (Andriluka et al. 2014) on the *MPII* dataset and report the PCKh@0.5 score.

Implementation Details The SHaRPose framework offers variability on three aspects: 1) the embedding size, which specifies the number of features carried by each token; 2) the parameter α, which determines the proportion of image patches utilized for generating high-resolution representations; 3) the threshold of the predicted pose quality Q_{thres}, which controls the number of samples that enter the fine stage. In this paper, we instantiate SHaRPose with two different sizes by scaling the embedding size. Other configurations like the depth (the number of Transformer blocks) are set the same. The detailed configurations of the instantiated SHaRPose models are presented in Table.2.

Training Details To ensure a fair comparison, all experiments presented in this paper are conducted using the MMPose framework (SenseTime 2020) on four NVIDIA RTX 3090 GPUs. The default data pipelines of MMPose are utilized. The masked autoencoder pretrain (He et al. 2022) is used as in (Xu et al. 2022b) for the purpose of exploring the potential capabilities of pure Transformers. UDP (Huang et al. 2020) is used for post-processing. The model is trained for 210 epochs with a learning rate of 5e-4, which is decreased to 5e-5 and 5e-6 at the 170th and 200th epochs, respectively. In particular, we aim to predict the confidence values as accurately as possible with the quality predictor. Because the convergence rate of the quality predictor is much faster than that of the heatmap (Zhao et al. 2021), we set $\lambda = 0$ in the first 180 epochs and $\lambda = 0.03$ in the subsequent epochs based on empirical analysis.

Results

Comparison to state-of-the-art methods on *COCO* We compare the performance and efficiency of our proposed method with several state-of-the-art (SOTA) approaches.

Validation set As shown in Table.3, under the input resolution 256×192, our SHaRPose-Small model achieves an AP of 74.2, which is a significant improvement of +8.6 AP over the TokenPose-T model and a +0.4 AP improvement over ViTPose-Small, while maintaining a faster inference speed. Furthermore, our SHaRPose-Base model achieves an AP of 75.5, which is a +0.4 AP improvement over HRNet-W48 and is $1.9\times$ faster than it. Notably, our model also demonstrates faster inference speed than TokenPose-L/D6, HRFormer, and ViTPose-Base, with comparable accuracy. At the higher input resolution of 384×288, our model's advantages become even more pronounced. The SHaRPose-Base model achieves a SOTA performance of 77.4 AP while maintaining lower GLOPs and higher throughput compared to other methods.

Method	Input	AP	AP^{50}	AP^{75}	AP^{L}	AP^{M}	AR	FPS ↑	GFLOPs ↓
TokenPose-T(Li et al. 2021b)†	256×192	65.6	86.4	73.0	71.5	63.1	72.1	348.1	**1.2**
ViTPose-Small(Xu et al. 2022b)†	256×192	73.8	90.3	81.3	75.8	67.1	79.1	360.3	5.7
SHaRPose-Small†	256×192	**74.2**↑0.4	90.2	**81.8**	**80.3**	**71.2**	**79.5**	**498.3**↑38%	4.9↓14%
SimpleBaseline(Xiao, Wu, and Wei 2018)	256×192	73.6	90.4	81.8	80.1	70.1	79.1	195.1	12.8
HRNet-W48(Wang et al. 2021a)	256×192	75.1	90.6	82.2	81.8	71.5	80.4	193.5	15.8
HRFormer-Base(Yuan et al. 2021)	256×192	75.6	90.8	82.8	82.6	71.7	80.8	122.3	14.7
TokenPose-L/D6(Li et al. 2021b)	256×192	75.4	90.0	81.8	82.4	71.8	80.4	348.2	**9.9**
ViTPose-Base(Xu et al. 2022b)	256×192	75.8	90.7	83.2	78.4	68.7	81.1	340.2	18.6
SHaRPose-Base	256×192	75.5	90.6	82.3	82.2	**72.2**	80.8	**392.8**↑15%	17.1↓10%
HRNet-W48(Wang et al. 2021a)	384×288	76.3	90.8	82.9	83.4	72.3	81.2	152.3	35.5
ViTPose-Base(Xu et al. 2022b)	384×288	76.9	90.9	83.2	83.9	73.1	82.1	143.3	44.1
SHaRPose-Small†	384×288	75.2	90.8	83.0	81.2	72.0	80.9	**395.3**	**11.0**
SHaRPose-Base	384×288	**77.4**↑0.5	**91.0**	**84.1**	**84.2**	**73.7**	**82.4**	196.6↑37%	32.9↓25%

Table 3: Comparison on *COCO* validation set. The same detection results with $56AP$ are used for human instances. No extra training data is involved for all results. The FPS(frame-per-second) is evaluated under an identical environment. † denotes the small-scale models. The underlined numbers emphasize the compared results. The best results are highlighted in bold.

Methods	Input	AP	AP^{50}	AP^{75}	AP^{L}	AP^{M}	AR	FPS ↑	GFLOPs ↓
SimpleBaseline(Xiao, Wu, and Wei 2018)	384×288	73.7	91.9	81.1	70.3	80.0	79.0	153.5	28.7
UDP-HRNet-W48(Huang et al. 2020)	384×288	76.5	92.7	84.0	73.0	82.4	81.6	152.3	35.5
DARK-HRNet-W48(Zhang et al. 2020)	384×288	76.2	92.5	83.6	72.5	82.4	81.1	150.4	32.9
TokenPose-L/D24(Li et al. 2021b)	384×288	75.9	92.3	83.4	72.2	82.1	80.8	117.2	**22.1**
ViTPose-Base(Xu et al. 2022b)	384×288	76.2	92.7	83.7	72.6	82.3	81.3	143.3	44.1
SHaRPose-Base	384×288	**76.7**↑0.5	**92.8**	**84.4**	**73.2**	**82.6**	**81.6**	**196.6**↑37%	32.9↓25%

Table 4: Comparison on *COCO* test-dev set, same detection results with $60.9AP$ is used for human instaces. We only report single dataset training results at resolution 384×288.

Test-dev set Table.4 demonstrates the results of the SOTA methods on *COCO* test-dev. SHaRPose-Base with 384×288 as input achieves $76.7AP$. Compared to HRNet with UDP and DARK post-processing, our model achieves +0.2 AP and +0.5 AP higher accuracy and nearly 1.3x faster inference speed. Compared to ViTPose-Base, our model has a +0.5 AP improvement and nearly $1.4\times$ higher throughput.

Comparison to state-of-the-art methods on *MPII* The results on *MPII* test set evaluated by PCKh@0.5 are displayed in Table.5. The input resolution is 256×256, and the ground-truth bounding boxes are used by default. Our SHaRPose-Base model achieves a PCKh score of 91.4, outperforming other methods while also demonstrating 2-3 times higher throughput.

Model	Simple Baseline	HRNet W48	TokenPose L/D24	OKDHP	SHaRPose Base
Mean↑	89.0	90.1	90.2	90.6	**91.4**
FPS↑	66.9	47.1	65.5	-	**212.4**

Table 5: Comparison on *MPII* val set. SHaRPose demonstrates a significant advantage.

α	0.0	0.3	0.4	0.5	1.0
AP	68.4	74.8	75.5	75.5	75.7
GFLOPs	13.3	15.8	17.1	18.2	24.9

(a) 256×192

α	0.3	0.5
AP	77.4	77.5
GFLOPs	32.9	38.9

(b) 384×288

Table 6: The effect of α at different settings

Ablation Study

Influence of α The parameter α is crucial in controlling the sparsity level of the high-resolution representation, and it impacts the calculation consumed by the fine stage.

As shown in Table.6, for 256x192 input resolution, augmenting alpha from 0 to 0.4 can bring significant accuracy improvement, but a marginal gain is observed with subsequent increments. Thus, considering the balance of accuracy and efficiency, we set $\alpha = 0.4$. For 384x288 input resolution, increasing α from 0.3 to 0.5 has little effect on accuracy but significantly increases computational costs. Therefore, setting α to 0.3 is sufficient to achieve accurate results.

Effect of quality predictor To evaluate the impact of the quality predictor, we adjust the value of Q_{thres} based on the same SHaRPose-Base model on *COCO* dataset, using the ground-truth bounding boxes. Fig.6 illustrates the number of samples that terminate inference after the coarse stage and how the overall AP varies with different values of Q_{thres}. We observe that as the value of Q_{thres} decreases, the model tends to skip more samples in the fine stage, resulting in a decrease in AP, but the AP^{50} and AP^{75} only change a little. Therefore, the appropriate choice of Q_{thres} depends on the specific application scenario and the required level of ac-

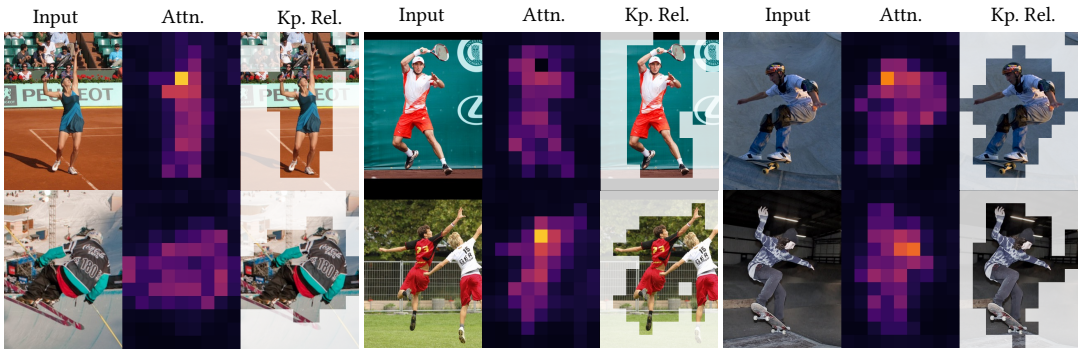

Figure 5: Visualization of keypoint-related regions. Three samples are chosen as examples. The first column gives the input image, the second column presents the accumulated attention map, and the third column shows the selected image regions.

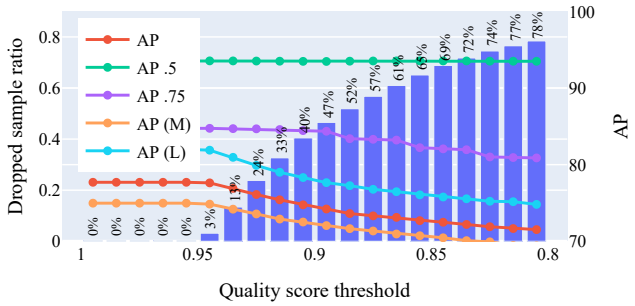

Figure 6: AP and the ratio of dropped samples on different settings of Q_{thres}. Each bar demonstrates the ratio of dropped samples with the Q_{thres} given, and the red line denotes the accuracy.

Method	AP	AP^{50}	FPS	GFLOPs
DynamicViT	71.8	89.4	182.8	12.8
EViT	72.7	90.1	424.1	12.8
SHaRPose-S	74.2	90.2	498.3	4.9
SHaRPose-B	75.5	90.6	392.8	17.1

Table 7: Comparison with pruning-based dynamic Transformers

curacy. In the experiments of section , we set $Q_{thres} = 0.95$ for comparison with other SOTA methods.

Necessity of the coarse-to-fine design To demonstrate the necessity of the coarse-to-fine architecture for pose estimation, we analyze from two perspectives: firstly, we perform comparative experiments on two pruning-based dynamic Transformers, namely DynamicViT(Rao et al. 2021) and EViT (Liang et al. 2022). We introduce keypoint tokens and employ a processing pipeline consistent with SHaRPose. As shown in Table.7, although EViT exhibits higher efficiency, its accuracy is compromised. This indicates that dynamic pruning in localization tasks limits the model's ability to generate precise outcomes. Secondly, we individually remove components of our framework, as shown in Table.8. The fine stage is indispensable for accuracy improvement, while the coarse stage, responsible for identifying keypoints-related image patches, plays a crucial role in reducing FLOPs.

Method	AP	AP^{50}	FPS	GFLOPs
Coarse-None	67.9	88.8	453.4	5.7
None-Fine	74.7	90.6	302.9	17.5
Coarse-Coarse	68.4	89.0	417.7	13.3
Coarse-NoSel-Fine	75.7	89.0	239.5	24.9
Coarse-Sel-Fine	75.5	90.6	392.8	17.1

Table 8: Comparison of different configurations of the proposed two stage framework

Visualization

Selected keypoint-related image patches Fig.5 presents some samples to visualize the keypoint-related regions. The second column exhibits the attention map that is accumulated between keypoint tokens and image patches, while the third column shows the keypoint-related regions, which are responsible for generating the high-resolution representation. It can be observed that the attention mechanism is primarily focused on the human instance, which aligns with the original design objective. Moreover, the attention intensity is particularly noticeable on the head since the *COCO* dataset contains more keypoints on the head.

Conclusion

In this paper, we provide an efficient pose estimation framework using only sparse high-resolution representations, named SHaRPose. Specifically, we introduce token-based keypoint representations into the coarse-to-fine framework to explicitly capture image parts that require high-resolution representations. In addition, we introduce a quality evaluation module, so that the model can quickly complete the inference of simple samples. Our quantitative experiments demonstrate the high accuracy and efficiency of our model. The visualization results also show the effectiveness of the proposed modules. This work provides directions for enhancing the computational efficiency of pose estimation methods using dynamic optimization strategies.

Acknowledgments

The authors would like to thank the editor and the anonymous reviewers for their critical and constructive comments and suggestions. This work was supported by the National Natural Science Fund of China under Grant No.62172222,62072354,62361166670, the Postdoctoral Innovative Talent Support Program of China under Grant 2020M681609, and the Fundamental Research Funds for the Central Universities under Grant QTZX23084.

References

Andriluka, M.; Pishchulin, L.; Gehler, P.; and Schiele, B. 2014. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In *CVPR*, 3686–3693.

Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.-R.; and Samek, W. 2015. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLoS ONE*, 10.

Badrinarayanan, V.; Kendall, A.; and Cipolla, R. 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12): 2481–2495.

Bolya, D.; Fu, C.-Y.; Dai, X.; Zhang, P.; Feichtenhofer, C.; and Hoffman, J. 2023. Token Merging: Your ViT but Faster. In *ICLR*.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-End Object Detection with Transformers. In *ECCV*, 213–229.

Chang, S.; Yuan, L.; Nie, X.; Huang, Z.; Zhou, Y.; Chen, Y.; Feng, J.; and Yan, S. 2020. Towards Accurate Human Pose Estimation in Videos of Crowded Scenes. In *ACMMM*, 4630–4634.

Chao, L.; Qiaoyong, Z.; Di, X.; and Shiliang, P. 2017. Skeleton-based action recognition with convolutional neural networks. In *ICMEW*, 597–600.

Chefer, H.; Gur, S.; and Wolf, L. 2021. Transformer Interpretability Beyond Attention Visualization. In *CVPR*, 782–791.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2018. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4): 834–848.

Chen, M.; Lin, M.; Li, K.; Shen, Y.; Wu, Y.; Chao, F.; and Ji, R. 2023. CF-ViT: A General Coarse-to-Fine Method for Vision Transformer. In *AAAI*, volume 37, 7042–7052.

Chen, Y.; Shen, C.; Wei, X.-S.; Liu, L.; and Yang, J. 2017. Adversarial PoseNet: A Structure-Aware Convolutional Network for Human Pose Estimation. In *ICCV*, 1221–1230.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Duan, H.; Zhao, Y.; Chen, K.; Lin, D.; and Dai, B. 2022. Revisiting Skeleton-based Action Recognition. In *CVPR*, 2959–2968.

Fang, H.-S.; Xie, S.; Tai, Y.-W.; and Lu, C. 2017. RMPE: Regional Multi-person Pose Estimation. In *ICCV*, 2353–2362.

Guo, S.; Liu, L.; Gan, Z.; Wang, Y.; Zhang, W.; Wang, C.; Jiang, G.; Zhang, W.; Yi, R.; Ma, L.; and Xu, K. 2022. ISDNet: Integrating Shallow and Deep Networks for Efficient Ultra-high Resolution Segmentation. In *CVPR*, 4351–4360.

Guo, W. 2020. Multi-Person Pose Estimation in Complex Physical Interactions. In *ACMMM*, 4752–4755.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked Autoencoders Are Scalable Vision Learners. In *CVPR*, 15979–15988.

Huang, J.; Zhu, Z.; Guo, F.; and Huang, G. 2020. The Devil Is in the Details: Delving Into Unbiased Data Processing for Human Pose Estimation. In *CVPR*, 5699–5708.

Jiang, T.; Lu, P.; Zhang, L.; Ma, N.; Han, R.; Lyu, C.; Li, Y.; and Chen, K. 2023. RTMPose: Real-Time Multi-Person Pose Estimation based on MMPose. arXiv:2303.07399.

Kawai, R.; Yoshida, N.; and Liu, J. 2022. Action Detection System Based on Pose Information. In *ACMMM Asia*, 40, 1–3.

Ke, L.; Danelljan, M.; Li, X.; Tai, Y.-W.; Tang, C.-K.; and Yu, F. 2022. Mask Transfiner for High-Quality Instance Segmentation. In *CVPR*, 4402–4411.

Khan, S.; Naseer, M.; Hayat, M.; Zamir, S. W.; Khan, F. S.; and Shah, M. 2022. Transformers in Vision: A Survey. *ACM Computing Surveys*, 54(10s).

Li, K.; Wang, S.; Zhang, X.; Xu, Y.; Xu, W.; and Tu, Z. 2021a. Pose Recognition with Cascade Transformers. In *CVPR*, 1944–1953.

Li, L.; Zhao, L.; Xu, L.; and Xu, J. 2022. Towards High Performance One-Stage Human Pose Estimation. In *ACMMM Asia*, 37, 1–5.

Li, Y.; Zhang, S.; Wang, Z.; Yang, S.; Yang, W.; Xia, S.-T.; and Zhou, E. 2021b. TokenPose: Learning Keypoint Tokens for Human Pose Estimation. In *ICCV*, 11293–11302.

Li, Z.; Ye, J.; Song, M.; Huang, Y.; and Pan, Z. 2021c. Online Knowledge Distillation for Efficient Pose Estimation. In *ICCV*, 11740–11750.

Liang, Y.; Ge, C.; Tong, Z.; Song, Y.; Wang, J.; and Xie, P. 2022. EViT: Expediting Vision Transformers via Token Reorganizations. In *ICLR*.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature Pyramid Networks for Object Detection. In *CVPR*, 936–944.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*, 740–755.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *ICCV*, 9992–10002.

Liu, Z.; Ning, J.; Cao, Y.; Wei, Y.; Zhang, Z.; Lin, S.; and Hu, H. 2022. Video Swin Transformer. In *CVPR*, 3192–3201.

Luo, X.; Huang, J.-B.; Szeliski, R.; Matzen, K.; and Kopf, J. 2020. Consistent Video Depth Estimation. *ACM Transactions on Graphics*, 39(4).

Mao, W.; Ge, Y.; Shen, C.; Tian, Z.; Wang, X.; and Wang, Z. 2021. TFPose: Direct Human Pose Estimation with Transformers. arXiv:2103.15320.

Newell, A.; Yang, K.; and Deng, J. 2016. Stacked Hourglass Networks for Human Pose Estimation. In *ECCV*, 483–499.

Niemirepo, T. T.; Viitanen, M.; and Vanne, J. 2020. Binocular Multi-CNN System for Real-Time 3D Pose Estimation. In *ACMMM*, 4553–4555.

Qiu, J.; Yan, X.; Wang, W.; Wei, W.; and Fang, K. 2022. Skeleton-Based Abnormal Behavior Detection Using Secure Partitioned Convolutional Neural Network Model. *IEEE Journal of Biomedical and Health Informatics*, 26(12): 5829–5840.

Ramakrishna, V.; Munoz, D.; Hebert, M.; Andrew Bagnell, J.; and Sheikh, Y. 2014. Pose Machines: Articulated Pose Estimation via Inference Machines. In *ECCV*, 33–47.

Rao, Y.; Zhao, W.; Liu, B.; Lu, J.; Zhou, J.; and Hsieh, C.-J. 2021. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In *NeurIPS*, volume 34, 13937–13949.

SenseTime. 2020. OpenMMLab Pose Estimation Toolbox and Benchmark. https://github.com/open-mmlab/mmpose. Accessed: 2023-03-20.

Shen, G.; Zhang, Y.; Li, J.; Wei, M.; Wang, Q.; Chen, G.; and Heng, P.-A. 2021. Learning Regularizer for Monocular Depth Estimation with Adversarial Guidance. In *ACMMM*, 5222–5230.

Tang, S.; Zhang, J.; Zhu, S.; and Tan, P. 2022. QuadTree Attention for Vision Transformers. In *ICLR*.

Tang, Y.; Zhao, L.; Yao, Z.; Gong, C.; and Yang, J. 2021. Graph-Based Motion Prediction for Abnormal Action Detection. In *ACMMM Asia*, 63, 1–7.

Tompson, J. J.; Jain, A.; LeCun, Y.; and Bregler, C. 2014. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. In *NeurIPS*, volume 27, 1799–1807.

Toshev, A.; and Szegedy, C. 2014. DeepPose: Human Pose Estimation via Deep Neural Networks. In *CVPR*, 1653–1660.

Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; Liu, W.; and Xiao, B. 2021a. Deep High-Resolution Representation Learning for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10): 3349–3364.

Wang, J.; and Torresani, L. 2022. Deformable Video Transformer. In *CVPR*, 14053–14062.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021b. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. In *ICCV*, 548–558.

Wei, S.-E.; Ramakrishna, V.; Kanade, T.; and Sheikh, Y. 2016. Convolutional Pose Machines. In *CVPR*, 4724–4732.

Xia, Z.; Pan, X.; Song, S.; Li, L. E.; and Huang, G. 2022. Vision Transformer with Deformable Attention. In *CVPR*, 4784–4793.

Xiao, B.; Wu, H.; and Wei, Y. 2018. Simple Baselines for Human Pose Estimation and Tracking. In *ECCV*, 472–487.

Xu, K.; Ye, F.; Zhong, Q.; and Xie, D. 2022a. Topology-Aware Convolutional Neural Network for Efficient Skeleton-Based Action Recognition. In *AAAI*, volume 36, 2866–2874.

Xu, Y.; Zhang, J.; ZHANG, Q.; and Tao, D. 2022b. ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation. In *NeurIPS*, volume 35, 38571–38584.

Yang, S.; Quan, Z.; Nie, M.; and Yang, W. 2021. TransPose: Keypoint Localization via Transformer. In *ICCV*, 11782–11792.

Yu, C.; Xiao, B.; Gao, C.; Yuan, L.; Zhang, L.; Sang, N.; and Wang, J. 2021. Lite-HRNet: A Lightweight High-Resolution Network. In *CVPR*, 10435–10445.

Yuan, Y.; Fu, R.; Huang, L.; Lin, W.; Zhang, C.; Chen, X.; and Wang, J. 2021. HRFormer: High-Resolution Vision Transformer for Dense Predict. In *NeurIPS*, volume 34, 7281–7293.

Zhang, C.; He, N.; Sun, Q.; Yin, X.; and Lu, K. 2021. Human Pose Estimation Based on Attention Multi-Resolution Network. In *ICMR*, 682–687.

Zhang, F.; Zhu, X.; Dai, H.; Ye, M.; and Zhu, C. 2020. Distribution-Aware Coordinate Representation for Human Pose Estimation. In *CVPR*, 7093–7102.

Zhang, F.; Zhu, X.; and Ye, M. 2019. Fast Human Pose Estimation. In *CVPR*, 3517–3526.

Zhao, L.; Xu, J.; Gong, C.; Yang, J.; Zuo, W.; and Gao, X. 2021. Learning to Acquire the Quality of Human Pose Estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(4): 1555–1568.