TokenPose: Learning Keypoint Tokens for Human Pose Estimation

Yanjie Li*[1,2] Shoukui Zhang[2] Zhicheng Wang[2] Sen Yang*[2,3]
Wankou Yang[3] Shu-Tao Xia†[1,4] Erjin Zhou[2]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University
[2]MEGVII Technology [3]Southeast University
[4]PCL Research Center of Networks and Communications, Peng Cheng Laboratory
lyj20@mails.tsinghua.edu.cn {zhangshoukui, wangzhicheng}@megvii.com
{yangsenius, wkyang}@seu.edu.cn xiast@sz.tsinghua.edu.cn zej@megvii.com

Abstract

Human pose estimation deeply relies on visual clues and anatomical constraints between parts to locate keypoints. Most existing CNN-based methods do well in visual representation, however, lacking in the ability to explicitly learn the constraint relationships between keypoints. In this paper, we propose a novel approach based on Token representation for human Pose estimation (TokenPose). In detail, each keypoint is explicitly embedded as a token to simultaneously learn constraint relationships and appearance cues from images. Extensive experiments show that the small and large TokenPose models are on par with state-of-the-art CNN-based counterparts while being more lightweight. Specifically, our TokenPose-S and TokenPose-L achieve 72.5 AP and 75.8 AP on COCO validation dataset respectively, with significant reduction in parameters (↓ 80.6% ; ↓ 56.8%) and GFLOPs (↓ 75.3%; ↓ 24.7%). Code is publicly available[1].

1. Introduction

2D human pose estimation aims to localize human anatomical keypoints which deeply relies on both visual cue and keypoints constraint relationships. It is a fundamental task in computer vision, which has attracted extensive attention from academia and industry.

Over the past decade, deep convolutional neural networks have achieved impressive performances on human pose estimation due to their powerful capacity in visual representation and recognition [8, 29, 22, 21, 38, 12, 37, 24]. Since *heatmap representation* has become the standard label representation to encode the positions of keypoints, most existing models tend to use fully convolutional layers to maintain the 2D-structure of feature maps until the network output. Nevertheless, there are usually no concrete variables abstracted by such CNN models to directly represent the keypoint entities, which limits the ability of the model to explicitly capture constraint relationships between parts.

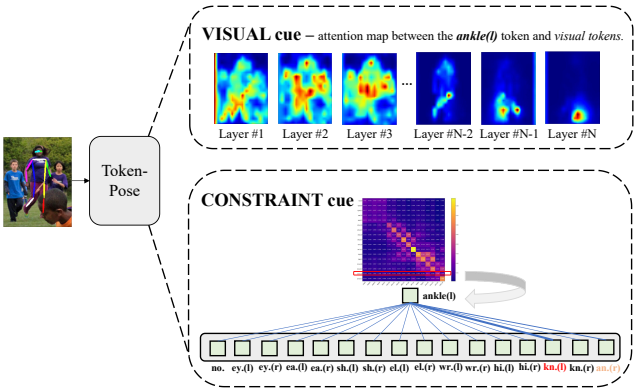

Figure 1. The process of predicting the location of the ***left ankle***. For visual cue learning, the proposed TokenPose focuses on the global context in the first few layers, and then gradually converges to some local regions as the network goes deeper. In the last few layers, TokenPose has considered *hip* and *knee* in turn which are close to the target keypoint, and finally localizes the position of the ***left ankle***. For constraint cue learning, TokenPose shows that localizing the ***left ankle*** mostly relies on the *left knee* and *right ankle*, corresponding to ***adjacency*** constraint and ***symmetric*** constraint respectively.

Recently, Transformer [35] and its variants that originated from natural language processing (NLP) have merged as new choices for various vision tasks. Its ability of modeling global dependencies is more powerful than CNN, which points out a promising way to efficiently capture relation-

*This work was done when Yanjie and Sen Yang were interns at MEGVII Tech.
†Corresponding author.
[1]https://github.com/leeyegy/TokenPose

ships between visual entities/elements. And in the field of NLP, all language elements such as words or characters are usually symbolized by embeddings or token vectors with fixed dimensions, so as to better measure their similarities in a vector space, like the way of word2vec [20].

We borrow such a concept of "token" and present a novel token-based representation for human pose estimation, namely TokenPose. Specifically, we conduct two different types of tokenizations: keypoint tokens and visual tokens. Visual tokens are yielded by uniformly splitting an image into patches and mapping the flattened patches into embeddings with fixed dimensions. Meanwhile, keypoint tokens are randomly initialized embeddings, each of which represents a specific type of keypoint (e.g., left knee, left ankle, right eye, etc.). The resulting keypoint tokens can learn both visual clues and constraint relations from interactions with visual tokens and the other keypoint tokens respectively. An example of how the proposed model predicts the location of *left ankle* is shown in Figure 1. The positions of keypoints are finally estimated over the token-based representation outputted by our network. The architecture of TokenPose is illustrated in Figure 2.

It is worth noting that TokenPose learns the statistic constraint relationships between keypoints from large amounts of data. Such information is encoded into keypoint tokens that can record their relationships by vector similarities. During inference, TokenPose associates keypoint tokens with those visual tokens whose corresponding patches possibly contain the target keypoints. By visualizing the attentions, we can observe how they interact and how the model exploits cues to localize keypoints.

The contributions are summarized as follows:

- We propose to use *token* to represent each *keypoint entity*. In this way, visual cue and constraint cue learning are explicitly incorporated into a unified framework.
- Both hybrid and pure Transformer-based architectures are explored in this work. As far as we know, proposed TokenPose-T is the first pure Transformer-based model for 2D human pose estimation.
- We conduct experiments over two widely-used benchmark datasets: COCO keypoint detection dataset [19] and MPII Human Pose dataset [1]. TokenPose achieves competitive state-of-the-art performance with much fewer parameters and computation cost compared with existing CNN-based counterparts.

2. Related Work

2.1. Human Pose Estimation

Deep convolutional neural networks have been applied to human pose estimation which greatly boost the model performance [32, 13, 29, 38, 22, 17, 21, 4, 7].

Recent heatmap-based methods tend to improve performance by stacking deeper network architecture. Hourglass [22] stacks blocks to enhance the heatmap estimation quality. SimpleBaseline [38] designs a simple architecture by stacking transposed convolution layers and achieves impressive performances. HRNet [29] proposes to maintain high-resolution representation through the whole process in order to provide spatially precise heatmap estimation. However, it is still hard for convolutional neural networks to capture and model constraint relationships between keypoints, which are important for human pose estimation.

2.2. Vision Transformer

Transformer [35] adopts encoder-decoder architecture based on self-attention and feed-forward network, which achieves great success in NLP. Recently, Transformer-based models [11, 34, 5, 14, 44, 45, 9, 39, 6, 36, 41, 28] have also shown enormous potential in various vision tasks.

Detection. DETR [5] proposes a Transformer based architecture to handle object detection end-to-end, effectively eliminating the need for many hand-designed components. Deformable DETR [45] then proposes to make attention modules only attend to a small set of key sampling points around a reference, achieving better performance than DETR. UP-DETR [9] unsupervisedly pre-train DETR by design randomly cropped patches.

Classification. ViT [11] proposes a pure Transformer model with patch embedding representation, which is pre-trained on large amounts of data and then fine-tuned on ImageNet dataset. DeiT [34] introduces a distillation token to ViT to learn knowledge from a teacher network, to avoid the pre-training on a large dataset. Tokens2Token [41] progressively encodes image into tokens and models the local structure information to reduce the sequence length.

Human Pose Estimation. Recent several works [27, 15, 18, 39, 43, 28] introduce Transformer for human pose estimation. PoseFormer [43] introduces Transformer for 3D pose estimation, based on 2D pose sequences in video frames. TransPose [39] tends to utilize attention layers built in Transformer to reveal the long-range dependencies of the predicted keypoints. However, TransPose lacks the ability to directly model the constraint relationships between keypoints. In this work, we propose to explicitly represent keypoints as token embeddings. And then both visual clues and constraint relations are simultaneously learned through self-attention interactions.

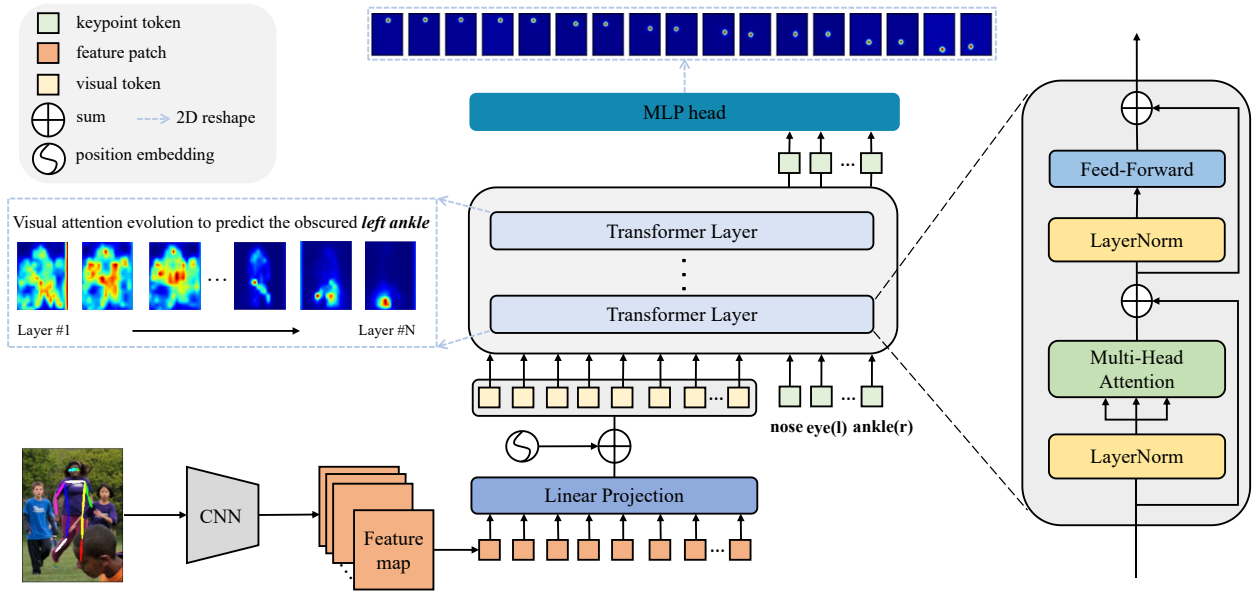

Figure 2. Schematic illustration of the proposed TokenPose. The feature maps extracted by CNN backbone are uniformly split into patches and flattened to 1D vectors. Visual tokens are yielded by adopting a linear projection to embed the flattened vectors. In addition, keypoint tokens are initialized randomly to represent each specific type of keypoint. Then, the 1D sequence of visual tokens and keypoint tokens are taken as input to Transformer encoder. Both appearance cues and anatomical constraint cues are captured through self-attention interactions in each Transformer layer. Finally, the keypoint tokens outputted by the last Transformer layer are used to predict the keypoints heatmaps via an MLP head.

3. Method

We firstly revisit the heatmap-based Fully Convolutional Networks (FCNs) for human pose estimation, and then describe our token-based design.

3.1. FCN-based Human pose estimation

The goal of human pose estimation is to localize N keypoints or parts from an image I with size $H \times W \times 3$. Nowadays, heatmap-based fully convolutional neural networks [37, 4, 22, 40, 7, 23, 38, 29] have been dominant solutions due to their high performance.

The widely-adopted pipeline is to utilize convolutional neural network to yield multi-resolution image feature maps, and a regressor to estimate N heatmaps of size $\hat{H} \times \hat{W}$. In order to yield N heatmaps, a 1×1 convolutional layer tends to be adopted to quickly adapt the channels of feature maps to N.

Despite the great success existing FCN-based methods have achieved, it's tough for CNN to explicitly capture constraint relationships between keypoints, which results in sub-optimal model design for this task.

3.2. Token-based Keypoint Representation

Visual tokens. The standard Transformer [35] accepts a 1D sequence of token embeddings as input. To handle 2D images, we follow the process of ViT [11]. An image $x \in \mathbb{R}^{H \times W \times C}$ is divided into a grid of $\frac{H}{P_h} \times \frac{W}{P_w}$ patches uniformly of size $P_h \times P_w$. And then each patch p is flattened into a 1D vector with size of $P_h \cdot P_w \cdot C$. To obtain a visual token v, each flattened patch p is then mapped into a d-dimensional embedding by a linear projection function $f : p \longrightarrow v \in \mathbb{R}^d$.

Considering human pose estimation is a location-sensitive vision task, 2D position embedding [35] pe_i is added to every specific visual token v_i to produce the input visual tokens $\texttt{[visual]} = \{v_1 + pe_1, v_2 + pe_2, ..., v_L + pe_L\}$, where $L = \frac{H \times W}{P_h \times P_w}$ is the amount of visual tokens . In this way, each visual token is yielded to represent a specific area of original image.

Keypoint tokens. We prepend N learnable d-dimensional embedding vectors to represent N target keypoints. We symbolize the keypoint tokens as `[keypoint]`. Together with visual tokens processed from image patches, keypoint tokens are accepted as the input of the Transformer. The state of N keypoint tokens at the output of the Transformer encoder serves as the N keypoints representation.

Transformer. Given the 1D token embeddings sequence $T = \{\texttt{[visual]}, \texttt{[keypoint]}\}$ as input, the Transformer encoder [35] learns keypoint feature representation by stacking M blocks. Each block contains a Multi-head Self-attention (MSA) module and a Multilayer Perceptron

Model	CNN backbone	Layers	Embedding size	Heads	Patch size	#Params	GFLOPs
TokenPose-Tiny	-	12	192	16	16×12	5.8M	1.3
TokenPose-Small-v1	stem-net	12	192	8	4×3	6.6M	2.2
TokenPose-Small-v2	stem-net	12	192	8	2×2	6.2M	11.6
TokenPose-Base	HRNet-W32-stage3	12	192	8	4×3	13.5M	5.7
TokenPose-Large/D6	HRNet-W48-stage3	6	192	8	4×3	20.8M	9.1
TokenPose-Large/D24	HRNet-W48-stage3	24	192	12	4×3	27.5M	11.0

Table 1. Architecture configurations. The model parameters and GFLOPs are computed under an image with 256×192 input resolution.

(MLP) module. In addition, layernorm (LN) is adopted before every module. Self-attention (SA) can be formulated as:

$$SA(T^{l-1}) = softmax(\frac{T^{l-1}W_Q(T^{l-1}W_K)^T}{\sqrt{d_h}})(T^{l-1}W_V) \quad (1)$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ are the learnable parameters of three linear projection layers, T^{l-1} is the output of the $(l-1)$-th layer, d is the dimension of tokens, and $d_h = d$. MSA is an extension of SA with h self-attention operations which are called "heads". In MSA, d_h is typically set to d/h.

$$MSA(\mathbf{T}) = [SA_1(T); SA_2(T); ...; SA_h(T)]W_P \quad (2)$$

where $W_P \in \mathbb{R}^{(h \cdot d_h) \times d}$. Note, the final heatmap prediction is based on the `[keypoint]` tokens outputted by the Transformer encoder with M blocks, which are denoted as $\{T_1^M, T_2^M, ..., T_N^M\}$.

Heatmap estimation. To obtain the 2D heatmaps with size of $\hat{H} \times \hat{W}$, the d-dimensional `[keypoint]` tokens outputted by the Transformer encoder are mapped into $\hat{H} \cdot \hat{W}$-dimensional feature vectors by linear projection. The mapped 1D vectors are then reshaped to 2D heatmaps. In addition, the MSE loss function is adopted to compare the predicted heatmaps and the groundtruth heatmaps.

Hybrid architecture. Instead of manipulating raw image patches directly, the input visual tokens can also be extracted from feature maps outputted by a convolution neural network [16]. In the hybrid architecture, CNN is adopted to extract low-level image features more efficiently.

4. Experiments

4.1. Model Variants

We provide both hybrid and pure Transformer-based variants for TokenPose. For hybrid architecture, convolutional neural networks with various depths are used for image feature extracting. The configuration details are presented in Table 1. Note, TokenPose-T* is the pure Transformer-based variant. TokenPose-S*, TokenPose-B and TokenPose-L* adopt stem-net[2], HRNet-W32 [29] and HRNet-W48 [29] as backbone, respectively.

In this paper, brief notation is used for convenience. For example, TokenPose-L/D24 means the "Large" variant with 24 Transformer layers. Unless noted otherwise, TokenPose-S and TokenPose-L are used as the abbreviations for TokenPose-Small-v2 and TokenPose-Large/D24.

4.2. COCO Keypoint Detection

Dataset. The COCO dataset [19] consists of more than $200,000$ images and $250,000$ person instances which are labeled with 17 keypoints. The COCO dataset is divided into *train/val/test-dev* sets, which contains 57k, 5k and 20k images respectively. All the experiments reported in this paper are trained only on the train2017 set. The methods are evaluated on the val2017 set and test-dev2017 set.

Evaluation metric. We adopt standard average precision (AP) as our evaluation metric on the COCO dataset. AP is calculated based on Object Keypoint Similarity (OKS): $OKS = \frac{\sum_i exp(-\hat{d}_i^2/2s^2k_i^2)\sigma(v_i>0)}{\sum_i \sigma(v_i>0)}$, where $\hat{d}_i$ is the Euclidean distance between the i-th predicted keypoint coordinate and the corresponding groundtruth, v_i is the visibility flag of the keypoint, s is the object scale, and k_i is a keypoint-specific constant.

Baseline settings. For model training, we use the Adam optimizer. For HRNet [29] and SimpleBaseline [38], we simply follow the original settings in their paper.

Implementation details. In this paper, we follow the two-stage top-down human pose estimation paradigm similar to [29, 7, 38, 25]. In the paradigm, the single person instance is firstly detected by a person detector, and then keypoints are predicted. We adopt the widely-used person detectors provided by SimpleBaseline [38] on the validation set and test-dev set. To alleviate the quantisation error, the well-designed coordinate decoding strategy [42] is adopted.

For our work, the base learning rate is set as 1e-3, and is dropped to 1e-4 and 1e-5 at the 200th and 260th epochs,

[2]It's widely used to quickly downsample the feature map into $1/4$ input resolution, consisting of a very shallow convolutional structure[29, 8].

Method	Pretrain		Input size	#Params	GFLOPs	gtbbox AP	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
	CNN	Transformer										
SimpleBaseline-Res50 [38]	Y	-	256×192	34.0M†	8.9†	72.4	70.4	88.6	78.3	67.1	77.2	76.3
SimpleBaseline-Res101 [38]	Y	-	256×192	53.0M	12.4	-	71.4	89.3	79.3	68.1	78.1	77.1
SimpleBaseline-Res152 [38]	Y	-	256×192	68.6M‡	15.7‡	74.3	72.0	89.3	79.8	68.7	78.9	77.8
HRNet-W32[29]	Y	-	256×192	28.5M§	7.1§	76.5	74.4	90.5	81.9	70.8	81.0	79.8
HRNet-W48 [29]	Y	-	256×192	63.6M⋔	14.6⋔	77.1	75.1	**90.6**	82.2	71.5	81.8	80.4
TokenPose-T (pure Transformer)	-	N	256×192	5.8M	1.3	-	65.6	86.4	73.0	63.1	71.5	72.1
TokenPose-S-v1	N	N	256×192	6.6M† (↓ 80.6%)	2.2† (↓ 75.3%)	75.0	72.5	89.3	79.7	68.8	79.6	78.0
TokenPose-S-v2	N	N	256×192	6.2M‡ (↓ 91.0%)	11.6‡ (↓ 23.7%)	76.1	73.5	89.4	80.3	69.8	80.5	78.7
TokenPose-B	Y	N	256×192	13.5M§ (↓ 52.6%)	5.7§ (↓ 19.7%)	-	74.7	89.8	81.4	71.3	81.4	80.0
TokenPose-L/D6	Y	N	256×192	20.8M⋔ (↓ 67.3%)	9.1⋔ (↓ 37.7%)	77.7	75.4	90.0	81.8	71.8	82.4	80.4
TokenPose-L/D24	Y	N	256×192	27.5M⋔ (↓ 56.8%)	11.0⋔ (↓ 24.7%)	**78.2**	**75.8**	90.3	**82.5**	**72.3**	**82.7**	**80.9**

Table 2. Comparisons on the COCO validation set, provided with the same detected human boxes. Pretrain means pre-training the corresponding parts on the ImageNet classification task. TokenPose-S*, TokenPose-B* and TokenPose-L* achieve competitive results to SimpleBaseline [38] and HRNet [29] respectively, with much fewer parameters&GFLOPs. We compute the percentages in terms of parameters&GFLOPs reduction between models marked with the same symbol.

Model	Embedding size	Layer	AP	#Params
TokenPose-L/D12	192	12	75.3	23.0M
TokenPose-L/D24	192	24	75.8	27.5M
TokenPose-L+/D12	384	12	75.5	38.2M

Table 3. Results of model scaling on the COCO validation set. The input image size is 256×192.

respectively. The total training process requires 300 epochs, given Transformer structure tends to rely on longer training to convergence. We follow the data augmentation in [29].

Comparison with state-of-the-art methods. As Table 2 shown, our proposed TokenPose achieves competitive performance compared with the other state-of-the-art methods via much fewer model parameters and GFLOPs. Compared to SimpleBaseline [38] that adopts ResNet-50 as the backbone, our TokenPose-S-v1 improves AP by 2.1 points with significant reduction in both model parameters (↓ 80.6%) and GFLOPs (↓ 75.3%). Compared to SimpleBaseline [38] that uses ResNet-152 as the backbone, our TokenPose-S-v2 achieves better performance, while using only 9.0% model parameters. Compared with HRNet-W32 and HRNet-W48, TokenPose-B and TokenPose-L achieve similar results with less than 50% model parameters, respectively. Besides, TokenPose-T obtains 65.6 AP with only 5.8M model parameters and 1.3 GFLOPs, without any convolution layer. Note, all Transformer parts are trained from scratch, without any pre-training. Also, Table 5 shows the results of our method and the existing state-of-the-art methods on the COCO test-dev set. With 384×288 as the input resolution, our TokenPose-L/D24 achieves 75.9 AP.

4.3. MPII Human Pose Estimation

Dataset & Evaluation metric. The MPII Human Pose dataset [1] contains images with full-body pose annotations

Model	Hea	Sho	Elb	Wri	Hip	Kne	Ank	Mean	#Params
SimpleBaseline-Res50 [38]	96.4	95.3	89.0	83.2	88.4	84.0	79.6	88.5	34.0M
SimpleBaseline-Res101 [38]	96.9	95.9	89.5	84.4	88.4	84.5	80.7	89.1	53.0M
SimpleBaseline-Res152 [38]	97.0	95.9	90.0	85.0	89.2	85.3	81.3	89.6	68.6M
HRNet-W32 [29]	96.9	**96.0**	90.6	85.8	88.7	86.6	82.6	90.1	28.5M
TokenPose-L/D6	97.1	95.9	**91.0**	85.8	**89.5**	86.1	**82.7**	**90.2**	**21.4M**
TokenPose-L/D12	**97.2**	95.8	90.7	85.9	89.2	86.2	82.3	90.1	23.5M
TokenPose-L/D24	97.1	95.9	90.4	**86.0**	89.3	**87.1**	82.5	**90.2**	28.1M

Table 4. Results on the MPII validation set (PCKh@0.5). The input size is 256×256.

obtained from various real-world activities. There are 40k person samples with 16 joints labels in the MPII dataset. In addition, the data augmentation is the same to that on the COCO dataset, except that the input images are cropped to 256×256. The head-normalized probability of correct keypoint (PCKh) [1] score is adopted for evaluation.

Results on the validation set. We follow the testing procedure in HRNet [29]. The PCKh@0.5 results of some top-performed methods are presented in Table 4. All the experiments are conducted with the input image size 256×256. It's shown that our proposed TokenPose achieves competitive performance while being more lightweight.

4.4. Ablation Study

Keypoint token fusion. Intermediate supervision is widely-used to help model training and improve the heatmap estimation quality especially when networks become very deep [22, 37, 33, 2]. Similarly, we propose to concatenate keypoint tokens outputted by different layers of the Transformer encoder correspondingly, namely 'keypoint token fusion', to help model training.

Taking TokenPose-L+/D12 with 12 Transformer layers as an example, the keypoint tokens output in the 4th, 8th and 12th layers are concatenated correspondingly. The resulting three times longer keypoint tokens are then sent into the

Method	Input size	#Params	GFLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
G-RMI [24]	353 × 257	42.6M	57	64.9	85.5	71.3	62.3	70	69.7
Integral Pose Regression [30]	256 × 256	45.0M	11	67.8	88.2	74.8	63.9	74	-
CPN [7]	384 × 288	-	-	72.1	91.4	80	68.7	77.2	78.5
RMPE [12]	320 × 256	28.1M	26.7	72.3	89.2	79.1	68	78.6	-
SimpleBaseline-Res152[38]	384 × 288	68.6M	35.6	73.7	91.9	81.1	70.3	80	79
HRNet-W48 [29]	256 × 192	63.6M	14.6	74.2	92.4	82.4	70.9	79.7	79.5
HRNet-W32 [29]	384 × 288	28.5M	16	74.9	92.5	82.8	71.3	80.9	80.1
TransPose-H-A6 [39]	256 × 192	17.5M	21.8	75.0	92.2	82.3	71.3	81.1	80.1
HRNet-W48 [29]	384 × 288	63.6M	32.9	75.5	**92.5**	83.3	71.9	81.5	80.5
TokenPose-S-v2	256 × 192	6.2M	11.6	73.1	91.4	80.7	69.7	79.0	78.3
TokenPose-B	256 × 192	13.5M	5.7	74.0	91.9	81.5	70.6	79.8	79.1
TokenPose-L/D6	256 × 192	20.8M	9.1	74.9	92.1	82.4	71.5	80.9	80.0
TokenPose-L/D24	256 × 192	27.5M	11.0	75.1	92.1	82.5	71.7	81.1	80.2
TokenPose-L/D24	384 × 288	29.8M	22.1	**75.9**	92.3	**83.4**	**72.2**	**82.1**	**80.8**

Table 5. Comparisons with state-of-the-art CNN-based models on the COCO test-dev set.

Model	Token fusion	AP	#Params
TokenPose-S	✗	73.5	6.2M
TokenPose-S	✓	72.6	6.7M
TokenPose-L+/D12	✗	75.3	35.8M
TokenPose-L+/D12	✓	75.5	38.2M

Table 6. The effects of keypoint token fusion for different models. The input image size is 256 × 192.

Position embedding	#Params	GFLOPs	AP	AR
✗	6.62M	2.07	67.0	73.4
Learnable	6.67M	2.23	71.4	77.1
2D sine	6.67M	2.23	72.5	78.0

Table 7. Results for various positional encoding strategies for TokenPose-S-v1. The input image size is 256 × 192.

MLP head to obtain the final heatmaps.

We report the results of TokenPose-S and TokenPose-L+/D12 with and without keypoint token fusion in Table 6. For TokenPose-L+/D12, using keypoint token fusion improves the result by 0.2 AP. However, for small variant like TokenPose-S, it causes performance degradation instead.

For TokenPose-Large with keypoint token fusion, we find the lower Transformer layers provide more meaningful evidence than the higher layers to understand the interaction process. We attribute this to the token fusion, which enables the final keypoint representation to directly exploit the information from the early layers. And such a phenomenon does not appear in the TokenPose-Small model without token fusion, in which the attention interactions progressively show clear and meaningful attention process. We will further describe it in Sec. 4.5.

Note that keypoint token fusion is only used in TokenPose-L given its very deep and complex structure.

Position embedding. Keypoint localization is a position-sensitive vision task. To illustrate the effect of position embedding, we conduct experiments based on TokenPose-S-v1 with different position embedding types (i.e., no position embedding, 2D sine and learnable position embedding). As Table 7 shown, employing position embedding significantly improves the performance by 5.5 AP at most. In particular, 2D sine position embedding performs better than learnable position embedding, which is as expected since the 2D spatial information is required for predicting heatmaps.

Scaling. Model scaling is a widely-used method to boost model performance, including width-wise [35, 10] scaling and depth-wise scaling [3, 26]. As shown in Table 3, both increasing depth and width help improve the results.

4.5. Visualization

To illustrate how the proposed TokenPose explicitly utilizes visual cue and constraint cue between parts to localize keypoints, we visualize the details during inference. We observe that a single model has similar behaviors for most common examples. We randomly choose some samples from the COCO validation set and visualize the details in Figure 3 and Figure 5.

Appearance cue. We visualize the attention maps between keypoint tokens and visual tokens of different Transformer layers in Figure 3. The attention maps are formed based on the attention scores between keypoint tokens and visual tokens. Note, we reshape the 1D sequence of attention scores according to their original space positions for the visualization.

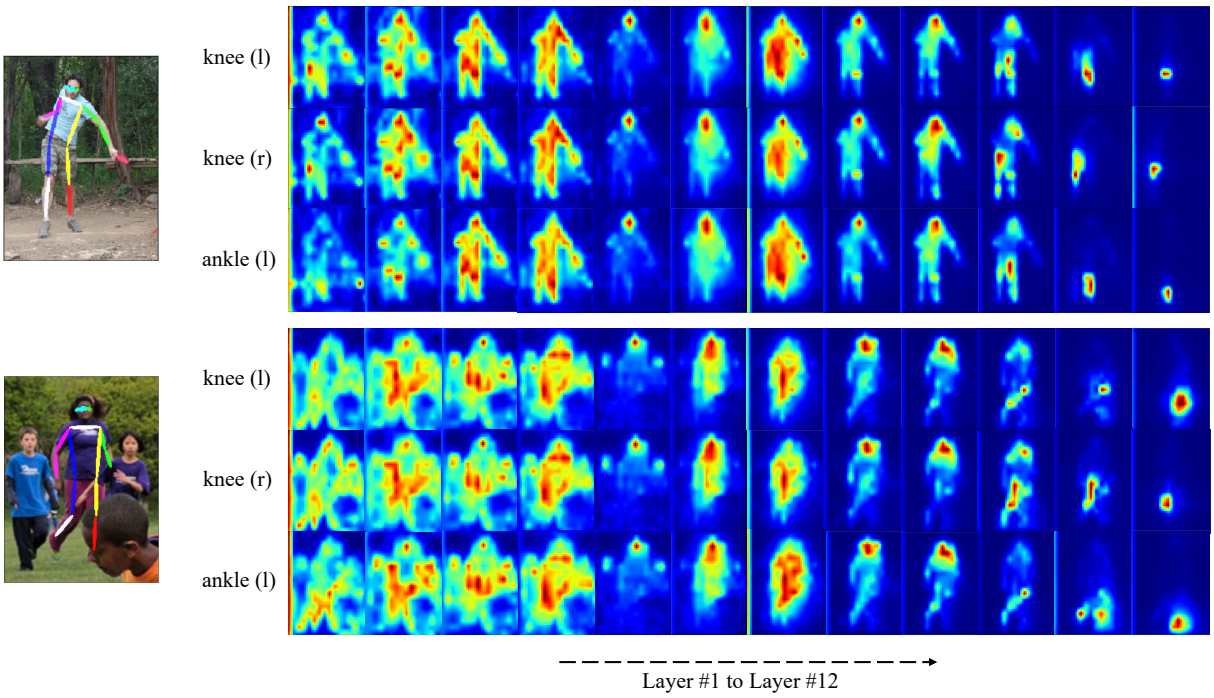

Figure 3. Visualization of the attention maps between keypoint tokens (e.g., *nose*, *elbow(l)*, and *elbow(r)*, etc.) and visual tokens in different layers of TokenPose-S, which consists of 12 Transformer layers. Note that we transform all visual token into its corresponding patch areas in the image. Redder color areas mean that the given type of keypoint has higher attention at these patches/visual tokens. The examples shown above and below are non-occluded and occluded cases, respectively.

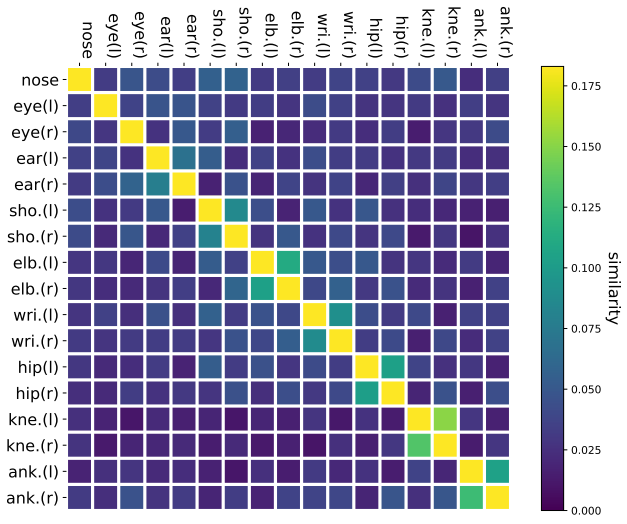

Figure 4. The inner product matrix of the learned keypoint tokens. We take the keypoint tokens that are fed into the first Transformer layer, compute their inner product matrix, scale them by $\sqrt{d}$, and use softmax to normalize them at columns. Thus each row can represent the learned prior constraint relationships for a given type of keypoint with other ones.

We choose two images for comparisons in Figure 3. As we can see, with the layer depth increasing, what the keypoint tokens capture is gradually from the whole body appearance cues to more precise local part cues. In the first few layers, multiple crowded persons may simultaneously give appearance cues as interference, but the model can progressively attend to the target person. In the subsequent layers, different types of keypoint tokens attend to their adjacent keypoints and the joints with high confidence evidence.

When inferring the occluded keypoints, the model behaves differently. As shown in Figure 3, we notice that the occluded *left-ankle* keypoint token pays higher attention to its symmetric joint (i.e., *right-ankle*) to obtain more clues.

Keypoint	Constraint	
	Top-1	Top-2
left shoulder	left elbow (0.026)	right shoulder (0.012)
left hip	right hip (0.037)	left knee (0.037)
right ankle	right knee (0.023)	left ankle (0.014)
nose	left eye (0.016)	right eye (0.016)
right wrist	right elbow (0.012)	left wrist (0.011)

Table 8. Top-2 constraints with regard to some keypoints for a randomly chosen sample. The values in parentheses represent the attention scores obtained from the final self-attention layer.

Keypoint constraints cue. The attention maps of keypoint tokens in the 2nd, 4th, 6th, 8th, 10th and 12th self-

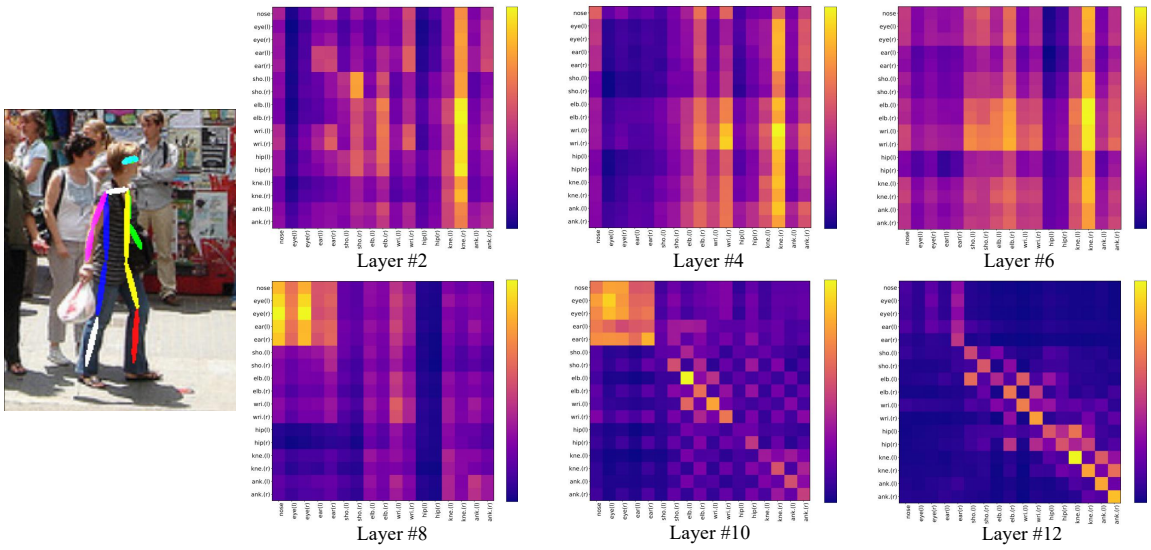

Figure 5. The attention interactions between keypoint tokens in the 2nd, 4th, 6th, 8th, 10th and 12th Transformer layers of TokenPose-S.

attention layers are visualized in Figure 5. In the first few layers, each keypoint pays attention to almost all other ones to construct global context. As the network goes deeper, each keypoint tends to mostly rely on several parts to yield the final prediction.

Specifically, we show top-2 constraints of some typical keypoints based on the final self-attention layer in Table 8. In particular, we observe that the top-2 constraints tends to be the *adjacent* and *symmetric constraint* of the target keypoint, which also conforms to the human visual system. For instance, predicting the *right wrist* mostly focuses on the constraints from the *right elbow* and *left wrist*, corresponding to its *adjacent* and *symmetric constraints* respectively.

Keypoint tokens learn prior knowledge from data. In proposed TokenPose, the input `[keypoint]` tokens which are taken as input to the first Transformer layer are totally learnable parameters. Such knowledge is related to the bias from the whole training dataset but independent of any specific image. During inference it will be exploited to facilitate the model to decode visual information from a concrete image and further make predictions.

We point out that such `[keypoint]` tokens act like object queries in DETR [5], in which each query slot finally has learned prior preference from data to specialize on certain areas and box sizes. In our settings, the input `[keypoint]` tokens learn statistical relevance between keypoints from the dataset, serving as prior knowledge.

To show what information is encoded in these input keypoint tokens, we calculate the inner product matrix of them. After being scaled and normalized, the matrix is visualized in Figure 4. We can see that one tends to be highly similar to its symmetric keypoints or adjacent keypoints. For instance, *left hip* is mostly related to *right hip* and *left shoulder* with similarity score 0.104 and 0.054 respectively. Such finding conforms to our common sense and reveals what the model learns. We also notice there is a work [31] which analyzes the statistic distributions between joints by computing the mutual information from MPII dataset annotation. In contrast, our model is able to automatically learn prior knowledge from training data and explicitly encode it in the input `[keypoint]` tokens.

5. Conclusion

In this paper, we propose a novel token-based presentation for human pose estimation, namely TokenPose. In particular, we split the image into patches to yield visual tokens and represent keypoint entities into token embeddings. This way, the proposed TokenPose is able to explicitly capture appearance cues and constraint cues by the self-attention interaction. We show that a low-capacity pure Transformer architecture without any pre-training can also work well. Besides, the hybrid architectures achieve competitive results compared to the state-of-the-art CNN-based methods at a much lower computational cost.

Acknowledgments

This paper is supported in part by the National Key R&D Plan of the Ministry of Science and Technology (Project No. 2020AAA0104400), and in part by the National Key Research and Development Program of China under Grant 2018YFB1800204, the National Natural Science Foundation of China under Grant 61771273, the R&D Program of Shenzhen under Grant JCYJ20180508152204044, and in part by the National Natural Science Foundation of China under Grant 61773117.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014. 2, 5

[2] Vasileios Belagiannis and Andrew Zisserman. Recurrent human pose estimation. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 468–475. IEEE, 2017. 5

[3] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 6

[4] Yuanhao Cai, Zhicheng Wang, Zhengxiong Luo, Binyi Yin, Angang Du, Haoqian Wang, Xinyu Zhou, Erjin Zhou, Xiangyu Zhang, and Jian Sun. Learning delicate local representations for multi-person pose estimation. In *ECCV*, 2020. 2, 3

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 2, 8

[6] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer, 2020. 2

[7] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7103–7112. IEEE Computer Society, 2018. 2, 3, 4, 6

[8] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5386–5395, 2020. 1, 4

[9] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. *arXiv preprint arXiv:2011.09094*, 2020. 2

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 6

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3

[12] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2334–2343, 2017. 1, 6

[13] Georgia Gkioxari, Alexander Toshev, and Navdeep Jaitly. Chained predictions using convolutional neural networks. In *European Conference on Computer Vision*, pages 728–743. Springer, 2016. 2

[14] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021. 2

[15] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7779–7788, 2020. 2

[16] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 4

[17] Ita Lifshitz, Ethan Fetaya, and Shimon Ullman. Human pose estimation using deep consensus voting. In *European Conference on Computer Vision*, pages 246–260. Springer, 2016. 2

[18] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. *arXiv preprint arXiv:2012.09760*, 2020. 2

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 4

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 2

[21] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in neural information processing systems*, pages 2277–2287, 2017. 1, 2

[22] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 1, 2, 3, 5

[23] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–286, 2018. 3

[24] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4903–4911, 2017. 1, 6

[25] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4903–4911, 2017. 4

[26] M Shoeybi, M Patwary, R Puri, P LeGresley, J Casper, and B Megatron-LM Catanzaro. Training multi-billion parameter language models using gpu model parallelism. *arXiv preprint arXiv:1909.08053*, 2019. 6

[27] Michael Snower, Asim Kadav, Farley Lai, and Hans Peter Graf. 15 keypoints is all you need. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6738–6748, 2020. 2

[28] Lucas Stoffl, Maxime Vidal, and Alexander Mathis. End-to-end trainable multi-instance pose estimation with transformers. *arXiv preprint arXiv:2103.12115*, 2021. 2

[29] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 1, 2, 3, 4, 5, 6

[30] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2018. 6

[31] Wei Tang and Ying Wu. Does learning specific features for related parts help human pose estimation? In *CVPR*, June 2019. 8

[32] Wei Tang, Pei Yu, and Ying Wu. Deeply learned compositional models for human pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 190–206, 2018. 2

[33] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 648–656, 2015. 5

[34] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 2

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 1, 2, 3, 6

[36] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021. 2

[37] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016. 1, 3, 5

[38] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018. 1, 2, 3, 4, 5, 6

[39] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Towards explainable human pose estimation by transformer. *arXiv preprint arXiv:2012.14214*, 2020. 2, 6

[40] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *proceedings of the IEEE international conference on computer vision*, pages 1281–1290, 2017. 3

[41] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021. 2

[42] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7093–7102, 2020. 4

[43] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. *arXiv preprint arXiv:2103.10455*, 2021. 2

[44] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020. 2

[45] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2