TransPose: Keypoint Localization via Transformer

Sen Yang  Zhibin Quan  Mu Nie  Wankou Yang*
School of Automation, Southeast University, Nanjing 210096, China
{yangsenius, 101101872, niemu, wkyang}@seu.edu.cn

Abstract

While CNN-based models have made remarkable progress on human pose estimation, what spatial dependencies they capture to localize keypoints remains unclear. In this work, we propose a model called TransPose, which introduces Transformer for human pose estimation. The attention layers built in Transformer enable our model to capture long-range relationships efficiently and also can reveal what dependencies the predicted keypoints rely on. To predict keypoint heatmaps, the last attention layer acts as an aggregator, which collects contributions from image clues and forms maximum positions of keypoints. Such a heatmap-based localization approach via Transformer conforms to the principle of Activation Maximization [19]. And the revealed dependencies are imagespecific and fine-grained, which also can provide evidence of how the model handles special cases, e.g., occlusion. The experiments show that TransPose achieves 75.8 AP and 75.0 AP on COCO validation and test-dev sets, while being more lightweight and faster than mainstream CNN architectures. The TransPose model also transfers very well on MPII benchmark, achieving superior performance on the test set when fine-tuned with small training costs. Code and pre-trained models are publicly available1.

1. Introduction

Deep convolutional neural networks have achieved impressive performances in the field of human pose estimation. DeepPose [56] is the early classic method, directly regressing the numerical coordinate locations of keypoints. Afterwards, fully convolutional networks like [60, 36, 38, 63, 12, 40, 61, 51] have become the mainstream by predicting keypoints heatmaps, which implicitly learn spatial dependencies between body parts. Yet, most prior works take deep CNN as a powerful black box predictor and focus on improving the network structure, what exactly happens inside the models or how they capture the spatial relationships between body parts remains unclear. However, from

*Corresponding author.
1https://github.com/yangsenius/TransPose

Figure 1. A schematic diagram of TransPose. **Below**: The inference pipeline. **Above**: Dependency areas for each predicted keypoint location. In this example, the person’s left-ankle is occluded by a dog. **Which exact image clues the model uses to infer the occluded joint?** The attention map (red box) gives fine-grained evidence beyond intuition: such a pose estimator highly relies on the image clues around the left ankle, left upper leg, and joints on the right leg to estimate the location of occluded left ankle.

the scientific and practical standpoints, the interpretability of the model can aid practitioners the ability to understand how the model associates structural variables to reach the final predictions and how a pose estimator handles various input images. It also can help model developers for debugging, decision-making, and further improving the design.

For existing pose estimators, some issues make it challenging to figure out their decision processes. (1) **Deepness**. The CNN-based models, such as [60, 38, 61, 51], are usually very deep non-linear models that hinder the interpretation of the function of each layer. (2) **Implicit relationships**. The global spatial relationships between body parts are implicitly encoded within the neuron activations and the weights of CNNs. It is not easy to decouple such relationships from large amounts of weights and activations in neural networks. And solely visualizing the intermediate features with a large number of channels (e.g. 256, 512 in SimpleBaseline architecture [61]) provides little meaning-
Stacking Many Convolutional Layers to Achieve a Large Receptive Field
Attend to Any Pairwise Locations by One Attention Layer

Figure 2. CNN vs. Attention. **Left:** The receptive filed enlarges in the deeper convolutional layer. **Right:** One self-attention layer can capture the pairwise relationship between any pair of locations.

ful explanations. (3) **Limited working memory in inferring various images.** The desired explanations for the model predictions should be image-specific and fine-grained. When inferring images, however, the static convolution kernels are limited in the ability to represent variables due to the limited working memory [23, 24, 27]. So it is difficult for CNNs to capture image-specific dependencies due to their content-independent parameters yet variable input image contents. (4) **Lack of tools.** Although there are already many visualization techniques based on gradient or attribution [19, 64, 49, 48, 21, 39, 69, 2], most of them focus on image classification rather than localization. They aim to reveal class-specific input patterns or saliency maps rather than to explain the relationships between structure variables (e.g., the locations of keypoints). By far, how to develop explainable pose estimators remains challenging.

In this work, we aim to build a human pose estimator that can explicitly capture and reveal the image-specific spatial dependencies between keypoints, as shown in Fig. 1. Due to the pool scaling property of convolution [45], we argue that convolution has advantages in extracting low-level features, but deeply stacking convolutions at high-level to enlarge the receptive field is not efficient to capture global dependencies. And such deepness increases the difficulty in interpreting CNN predictions. Transformer architecture [58] has a natural advantage over CNNs in terms of drawing pairwise or higher-order interactions. As shown in Fig. 2, attention layers enable the model to capture interactions between any pairwise locations, and its attention map acts as an immediate memory to store these dependencies.

Based on these considerations, we propose a novel model called TransPose, using convolutions to extract features at low-level and Transformer to capture global dependencies at high-level. In detail, we flatten the feature maps as input to Transformer and recover its output into the 2D-structure heatmaps. In such a design, the last attention layer in Transformer specially acts as an **aggregator**, which collects different contributions from all image locations by attention scores and finally forms the maximum positions in the heatmaps. This type of keypoint localization approach via Transformer establishes a connection with the interpretability of Activation Maximization [19, 49] and extends it to the localization task. The resulting attention scores can indicate what concrete image clues significantly contribute to the predicted locations. With such evidence, we can further analyze the behaviors of the model by examining the influence of different experimental variables. In summary, our contributions are as follow:

- We introduce Transformer for human pose estimation to predict heatmap-based keypoints positions, which can efficiently capture the spatial relationships between human body parts.
- We demonstrate that our keypoint localization approach based on Transformer conforms to the interpretability of Activation Maximization [19, 49]. Qualitative analysis reveals the dependencies beyond intuition, which are image-specific and fine-grained.
- TransPose models achieve competitive performances against state-of-the-arts CNN-based models via fewer parameters and faster speeds. TransPose achieves 75.8 AP and 75.0 AP on COCO validation set and test-dev set, with 73% fewer parameters and 1.4× faster than HRNet-W48. In addition, our model transfers very well on MPII benchmark.

2. Related Work

2.1. Human Pose Estimation

Deep CNNs have achieved great success in human pose estimation. The inductive biases of vanilla convolution kernel [31, 30] are locality and translation equivariance. It proves to be efficient to extract low-level image feature. For human pose estimation, capturing global dependencies is crucial [46, 55, 60, 40], but the locality nature of convolution makes it impossible to capture long-range interactions. A typical but brute solution is to enlarge the receptive field, e.g. by downsampling the resolution, increasing the depth or expanding the kernel size. Further, sophisticated strategies are proposed such as multi-scale fusion [38, 43, 63, 12, 51, 15, 13], stacking [60, 61, 38], or high-resolution representation [51]; meanwhile, many successful architectures have emerged such as CPM [60], Hourglass Network [38], FPN [63], CPN [12], SimpleBaseline [61], HRNet [51], RSN [8], even automated architectures [62, 22, 37, 14, 68]. But as the architecture is becoming more complex, it is more challenging but imperative than ever to seek the interpretability of human pose estimation models. In contrast, our model can estimate human pose in an efficient and explicit way.
2.2. Explainability

Explainability means a better understanding for human of how the model makes predictions. As surveyed by [47], many works define the goal for explanation is to determine what inputs are the most relevant to the prediction, which is also the goal we seek in this paper. [19, 32] perform gradient descent in the input space to find out what input patterns can maximize a given unit. [49, 20] further consider generating the image-specific class saliency maps. [64] uses DeConvNet to generate feature activities to show what convolutional layers have learned. Some pose estimation methods [32, 67] visualize the feature maps by choosing specific neurons or channels but the results fail to reveal the spatial relationship between parts. [54] estimates the probability distributions and mutual information between keypoints, yet only revealing the statistic information rather than image-specific explanations. There are also works like Network Dissection [3], Feature Visualization [39], Excitation Backprop [66], LRP attribution method [2], CAM [69], and Grad-CAM [48], which aim to explain the prediction of CNN classifier or visualize the saliency area significantly affecting the class. Different from most prior works, we aim to reveal the fine-grained spatial dependencies between body joints variables in the structural skeleton. And our model can directly exploit the attention patterns to holistically explain its predictions without the help of external tools. We also notice a recent paper [10] that develops LRP-based method to compute relevance to explain the predictions of Transformer. It takes ViT model [18] to visualize class-specific relevance map, showing reasonable results. Unlike their goal, we focus on revealing what clues contribute to visual keypoint localizations, and the attentions in our model provide clear evidence for the predictions.

It is worth noting that there are some works, such as CoordConv [35] and Zero Padding [29], to explain how the neural network predicts the positions and stores the position information by designing proxy tasks. We also conduct experiments to investigate the importance of position embedding for predicting the locations and its generalization on unseen input scales.

2.3. Transformer

Transformer was proposed by Vaswani et al. [58] for neural machine translation (NMT) task [53]. Large Transformer-based models like BERT [17], GPT-2 [44] are often pre-trained on large amounts of data and then fine-tuned for smaller datasets. Recently, Vision Transformer or attention-augmented layers have merged as new choices for vision tasks as [42, 45, 5, 18, 57, 9, 11, 16, 70, 59]. DETR [9] directly predicts a set of object instances by introducing object queries. ViT [18] is to pre-train a pure Transformer on large data and then fine-tuned on ImageNet for image classification. DeiT [57] introduces a distillation token to learn knowledge from a teacher. There are also works [26, 28, 33] applying Transformers to 3D pose estimation. [26] fuses features from multi-view images by attention mechanism. [28, 33] output 1D sequences composed of joint/vertex coordinates of pose. Unlike them, we use Transformer to predict the 2D heatmaps represented with spatial distributions of keypoints for 2D human pose estimation problem.

3. Method

Our goal is to build a model that can explicitly capture global dependencies between human body parts. We first describe the model architecture. Then we show how it exploits self-attention to capture global interactions and establish a connection between our method and the principle of Activation Maximization.

3.1. Architecture

As illustrated in Fig. 3, TransPose model consists of three components: a CNN backbone to extract low-level image feature; a Transformer Encoder to capture long-range spatial interactions between feature vectors across the locations; a head to predict the keypoints heatmaps.

**Backbone.** Many common CNNs can be taken as the backbone. For better comparisons, we choose two typical CNN architectures: ResNet [25] and HRNet [51]. We only retain the initial several parts of the original ImageNet pre-trained CNNs to extract feature from images. We name them ResNet-S and HRNet-S, the parameters numbers of which are only about 5.5% and 25% of the original CNNs.

**Transformer.** We follow the standard Transformer architecture [58] as closely as possible. And only the Encoder is employed, as we believe that the pure heatmaps prediction task is simply an encoding task, which compresses the original image information into a compact position representation of keypoints. Given an input image $I \in \mathbb{R}^{3 \times H \times W}$, we assume that the CNN backbone outputs a 2D spatial structure image feature $X_f \in \mathbb{R}^{d \times H \times W}$ whose feature dimension has been transformed to $d$ by a $1 \times 1$ convolution. Then, the image feature map is flattened into a sequence $X \in \mathbb{R}^{L \times d}$, i.e., $L$ $d$-dimensional feature vectors where $L = H \times W$. It goes through $N$ attention layers and feed-forward networks (FFNs).

**Head.** A head is attached to Transformer Encoder output $E \in \mathbb{R}^{L \times d}$ to predict $K$ types of keypoints heatmaps $P \in \mathbb{R}^{K \times H' \times W'}$ where $H', W' = H_f/4, W_f/4$ by default. We firstly reshape $E$ back to $d \times H' \times W'$ by default. Then we mainly use a $1 \times 1$ convolution to reduce the channel dimension of $E$ from $d$ to $K$. If $H, W$ are not equal $H, W'$, an additional bilinear interpolation or a $4 \times 4$ transposed convolution is used to do upsampling before $1 \times 1$ convolution. Note, a $1 \times 1$ convolution is completely equivalent to a position-wise linear transformation layer.
3.2. Resolution Settings.

Due to the computational complexity of per self-attention layer is \( O((HW)^2 \cdot d) \), we restrict the attention layers to operate at a resolution with \( r \times \) downsampling w.r.t. the original input, i.e., \( H, W = H_1/r, W_1/r \). In the common human pose estimation architectures [60, 38, 61, 51], \( 32 \times \) downsampling is usually adopted as a standard setting to obtain a very low-resolution map containing global information. In contrast, we adopt \( r = 8 \) and \( r = 4 \) setting for ResNet-S and HRNet-S, which are beneficial to the trade-off between the memory footprint for attention layers and the loss in detailed information. As a result, our model directly captures long-range interactions at a higher resolution, while preserving the fine-grained local feature information.

3.3. Attentions are the Dependencies of Localized Keypoints

**Self-Attention mechanism.** The core mechanism of Transformer [58] is multi-head self-attention. It first projects an input sequence \( X \in \mathbb{R}^{L \times d} \) into queries \( Q \in \mathbb{R}^{L \times d} \), keys \( K \in \mathbb{R}^{L \times d} \) and values \( V \in \mathbb{R}^{L \times d} \) by three matrices \( W_Q, W_K, W_V \in \mathbb{R}^{d \times d} \). Then, the attention scores matrix\(^2\) \( A \in \mathbb{R}^{N \times N} \) is computed by:

\[
A = \text{softmax} \left( \frac{QK^\top}{\sqrt{d}} \right).
\]  

Each query \( q_i \in \mathbb{R}^d \) of the token \( x_i \in \mathbb{R}^d \) (i.e., feature vector at location \( i \)) computes similarities with all the keys to achieve a weight vector \( w_i = A_{i,:} \in \mathbb{R}^{1 \times L} \), which determines how much dependency is needed from each token in the previous sequence. Then an increment is achieved by a linear sum of all elements in Value matrix \( V \) with the corresponding weight in \( w_i \) and added to \( x_i \). By doing this, the attention maps can be seen as dynamic weights that determined by specific image content, reweighting the information flow in the forward propagation.

Self-attention captures and reveals how much contribution the predictions aggregate from each image location. Such contributions from different image locations can be reflected by the gradient [49, 2, 48]. Therefore, we concretely analyze how \( x_j \) at image/sequence location \( j \) affects the activation \( h_i \) at location \( i \) the predicted keypoint heatmaps, by computing the derivative of \( h_i \in \mathbb{R}^K \) (K types of keypoints) w.r.t the \( x_j \) at location \( j \) of the input sequence of the last attention layer. And we further assume \( G := \frac{\partial h_i}{\partial x_j} \) as a function w.r.t. a given attention score \( A_{i,j} \). We obtain:

\[
G(A_{i,j}) \approx A_{i,j} \cdot W_f \cdot W_v^\top + W_f \cdot A_{i,j} \cdot K + B
\]

where \( K, B \in \mathbb{R}^{K \times d} \) are static weights (fixed when inferring) and shared across all image locations. The derivations of Eq. 2 are shown in supplementary. We can see that the function \( G \) is approximately linear with \( A_{i,j} \), i.e., the degrees of contribution to the prediction \( h_i \) directly depend on its attention scores at image locations.

Especially, the last attention layer acts as an aggregator, which collects contributions from all image locations according to attentions and forms the maximum activations in the predicted keypoint heatmaps. Although the layers in FFN and head cannot be ignored, they are position-wise, which means they approximately linearly transform the contributions from all locations by the same transformation without changing their relative proportions.

The activation maximum positions are the keypoints’ locations. The interpretability of Activation Maximization

---

\(^2\)Here we consider single-head self attention. For multi-head self-attention, the attention matrix is the average of attention maps in all heads.
Table 1. Architecture configurations for different TransPose models. More details about the backbones are described in supplementary.

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Backbone</th>
<th>Downsampling for Attention</th>
<th>Upsampling</th>
<th>#Layers</th>
<th>Heads</th>
<th>d</th>
<th>h</th>
<th>#Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransPose-R-A3*</td>
<td>ResNet-Small*</td>
<td>1/8</td>
<td>Bilinear Interpolation</td>
<td>3</td>
<td>8</td>
<td>256</td>
<td>512</td>
<td>5.0M</td>
</tr>
<tr>
<td>TransPose-R-A3</td>
<td>ResNet-Small</td>
<td>1/8</td>
<td>Deconvolution</td>
<td>3</td>
<td>8</td>
<td>256</td>
<td>1024</td>
<td>5.2M</td>
</tr>
<tr>
<td>TransPose-R-A4</td>
<td>ResNet-Small</td>
<td>1/8</td>
<td>Deconvolution</td>
<td>4</td>
<td>8</td>
<td>256</td>
<td>1024</td>
<td>6.0M</td>
</tr>
<tr>
<td>TransPose-H-S</td>
<td>HRNet-Small-W32</td>
<td>1/4</td>
<td>None</td>
<td>4</td>
<td>1</td>
<td>64</td>
<td>128</td>
<td>8.0M</td>
</tr>
<tr>
<td>TransPose-H-A4</td>
<td>HRNet-Small-W48</td>
<td>1/4</td>
<td>None</td>
<td>4</td>
<td>1</td>
<td>96</td>
<td>192</td>
<td>17.5M</td>
</tr>
<tr>
<td>TransPose-H-A6</td>
<td>HRNet-Small-W48</td>
<td>1/4</td>
<td>None</td>
<td>6</td>
<td>1</td>
<td>96</td>
<td>192</td>
<td>17.5M</td>
</tr>
</tbody>
</table>

Table 2. Results on COCO validation set, all provided with the same detected human boxes. TransPose-R-* and TransPose-H-* achieve competitive results to SimpleBaseline and HRNet, with fewer parameters and faster speeds. The reported FLOPs of SimpleBaseline and HRNet only include the convolution and linear layers.

<table>
<thead>
<tr>
<th>Method</th>
<th>Input Size</th>
<th>AP</th>
<th>AR</th>
<th>#Params</th>
<th>FLOPs</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimpleBaseline-Res50 [61]</td>
<td>256×192</td>
<td>70.4</td>
<td>76.3</td>
<td>34.0M</td>
<td>8.9G</td>
<td>114</td>
</tr>
<tr>
<td>SimpleBaseline-Res101 [61]</td>
<td>256×192</td>
<td>71.4</td>
<td>76.3</td>
<td>53.0M</td>
<td>12.4G</td>
<td>92</td>
</tr>
<tr>
<td>SimpleBaseline-Res152 [61]</td>
<td>256×192</td>
<td>72.0</td>
<td>77.8</td>
<td>68.6M</td>
<td>35.3G</td>
<td>62</td>
</tr>
<tr>
<td>TransPose-R-A3*</td>
<td>256×192</td>
<td>71.5</td>
<td>76.9</td>
<td>5.0M (↓85%)</td>
<td>5.4G</td>
<td>137 (↑20%)</td>
</tr>
<tr>
<td>TransPose-R-A3</td>
<td>256×192</td>
<td>71.7</td>
<td>77.1</td>
<td>5.2M (↓85%)</td>
<td>8.0G</td>
<td>141 (↑23%)</td>
</tr>
<tr>
<td>TransPose-R-A4</td>
<td>256×192</td>
<td>72.6</td>
<td>78.0</td>
<td>6.0M (↓82%)</td>
<td>8.9G</td>
<td>138 (↑21%)</td>
</tr>
<tr>
<td>HRNet-W32 [51]</td>
<td>256×192</td>
<td>74.4</td>
<td>79.8</td>
<td>28.5M</td>
<td>7.2G</td>
<td>28</td>
</tr>
<tr>
<td>HRNet-W48 [51]</td>
<td>256×192</td>
<td>75.1</td>
<td>80.4</td>
<td>63.6M</td>
<td>14.6G</td>
<td>27</td>
</tr>
<tr>
<td>TransPose-H-S</td>
<td>256×192</td>
<td>74.2</td>
<td>78.0</td>
<td>8.0M (↓72%)</td>
<td>10.2G</td>
<td>45 (↑61%)</td>
</tr>
<tr>
<td>TransPose-H-A4</td>
<td>256×192</td>
<td>75.3</td>
<td>80.3</td>
<td>17.3M (↓73%)</td>
<td>17.5G</td>
<td>41 (↑52%)</td>
</tr>
<tr>
<td>TransPose-H-A6</td>
<td>256×192</td>
<td><strong>75.8</strong></td>
<td><strong>80.8</strong></td>
<td>17.5M (↓73%)</td>
<td><strong>21.8G</strong></td>
<td><strong>38 (↑41%)</strong></td>
</tr>
</tbody>
</table>

(AM) [19, 49] lies in: the input region which can maximize a given neuron activation can explain what this activated neuron is looking for.

In this task, the learning target of TransPose is to expect the neuron activation $h_i$ at location $i^*$ of the heatmap to be maximally activated where $i^*$ represents the groundtruth location of a keypoint:

$$\theta^* = \arg \max_{\theta} \max_i h_i, (\theta, I). \quad (3)$$

Assuming the model has been optimized with parameters $\theta^*$ and it predicts the location of a particular keypoint as $i$ (maximum position in a heatmap), why the model predicts such prediction can be explained by the fact that those locations $J$, whose element $j$ has higher attention score ($\geq \delta$) with $i$, are the dependencies that significantly contribute to the prediction. The dependencies can be found by:

$$J = \{ j | A_{i,j}(\theta^*, I) \geq \delta \}, \quad (4)$$

where $A \in \mathbb{R}^{L \times L}$ is the attention map of the last attention layer and also a function w.r.t $\theta$ and $I$, i.e., $A = A(\theta^*, I)$. Given an image $I$ and a query location $i$, $A_{i,:}$ can reveal what dependencies a predicted location $i$ highly relies on, we define it dependency area. $A_{i,:}$ can reveal what area a location $j$ mostly affects, we define it affected area.

For the traditional CNN-based methods, they also use heatmap activations as the keypoint locations, but one cannot directly find the explainable patterns for the predictions due to the deepness and highly non-linearity of deep CNNs. The AM-based methods [19, 32, 64, 49] may provide insights while they require extra optimization costs to learn explainable patterns the convolutional kernels prefer to look for. Different from them, we extend AM to heatmap-based localization via Transformer, and we do not need extra optimization costs because the optimization has been implicitly accomplished in our training, i.e., $A = A(\theta^*, I)$. The defined dependency area is the pattern we seek, which can show image-specific and keypoint-specific dependencies.

4. Experiments

Dataset. We evaluate our models on COCO [34] and MPII [1] datasets. COCO contains 200k images in the wild and 250k person instances. Train2017 consists of 57k images and 150k person instances. Val2017 set contains 5k images and test-dev2017 consists of 20k images. In Sec 4.2, we show the experiments on MPII [1]. And we adopt the standard evaluation metrics of these benchmarks.

Technical details. We follow the top-down human pose estimation paradigm. The training samples are the cropped images with single person. We resize all input images into 256 × 192 resolution. We use the same training strategies, data augmentation and person detected results as [51]. We also adopt the coordinate decoding strategy proposed by [65] to reduce the quantisation error when decoding from downscaled heatmaps. The feed forward layers are trained with 0.1 dropout and ReLU activate function. Next, we name the models based on ResNet-S and HRNet-S TransPose-R and TransPose-H, abbreviated as TP-R and TP-H. The architecture details are reported in Tab. 1. We use Adam optimizer for all models. Training epochs are 230 for TP-R and 240 for TP-H. The cosine annealing learning rate decay is used. The learning rates for TP-R-A4 and
TP-H-A6 models decay from 0.0001 to 0.00001, we recommend using such a schedule for all models. Considering the compatibility with backbone and the memory consumption, we adjust the hyperparameters of Transformer encoder to make the model capacity not very large. In addition, we use 2D sine position embedding as the default position embedding. We describe it in the supplementary.

4.1. Results on COCO keypoint detection task

We compare TransPose with SimpleBaseline, HRNet, and DARK [65]. Specially, we trained the DARK-Res50 on our machines according to the official code with TransPose-R-A4’s data augmentation, we achieve 72.0 AP; when using the totally same data augmentation and long training schedule of TransPose-R-A4 for it, we obtain 72.1 AP (+0.1 AP). The other results showed in Tab. 2 come from the papers. We test all models on a single NVIDIA 2080Ti GPU with the same experimental conditions to compute the average FPS. Under the input resolution – 256 × 192, TransPose-R-A4 and TransPose-H-A6 have obviously over-performed SimpleBaseline-Res152 (+0.6 AP) [61], HRNet-W48 (+0.7 AP) [51] and DARK-HRNet [65] (+0.2 AP), with significantly fewer model parameters and faster speeds. Tab. 3 shows the results on COCO test set.

![Table 3. Comparisons with state-of-the-art CNN-based models on COCO test-dev set. Tested on smaller input resolution 256 × 192, our models achieve comparable performances with the others.](image)

4.2. Transfer to MPII benchmark

Typical pose estimation methods often separately train and evaluate their models on COCO and MPII [1]. Motivated by the success of pre-training in NLP and recent ViT [18], we try to transfer our pre-trained models to MPII. We replace the final layer of the pre-trained TransPose model with a uniform-initialized $d \times 16$ linear layer for MPII. When fine-tuning, the learning rates for the pre-trained and final layers are 1e-5 and 1e-4 with decay.

For comparisons, we fine-tune the pre-trained DARK-HRNet on MPII with the same settings, and train these models on MPII by standard full-training settings. As shown in Tab. 5 and Fig. 4, the results are interesting: even with large model capacity (28.5M), the improvement (+1.4 AP) brought by pre-training DARK-HRNet is smaller than pre-training (+2.0 AP). In Tab. 5 and Fig. 4, the results are interesting: even with large model capacity (28.5M), the improvement (+1.4 AP) brought by pre-training DARK-HRNet is smaller than pre-training (+2.0 AP). With 256 × 256 input resolution and fine-tuning on MPII train and val sets, the best result on MPII test set yielded by TransPose-H-A6 is 93.5% accuracy, as shown in Fig. 6. These results show that pre-training and fine-tuning could significantly reduce training costs and improve the performances, particularly for the pre-trained TransPose models.

**Discussion.** The pre-training and fine-tuning for Transformer-based models have shown favorable results in NLP [17, 44] and recent vision models [18, 11, 16]. Our initial results on MPII also suggest that training Transformer-based models on large-scale pose-related data may be a

![Figure 4. Performances on validation set when fine-tuning models (listed in Tab. 5) with different epochs on MPII training set.](image)

**Table 5. Fine-tuning and full-training performances on MPII validation set. ⊘ means full-training on MPII without COCO pre-training. ⊘⇒↑ means transferring the pretrained model and fine-tuning on MPII; adding ↑ means fine-tuning MPII on input resolution 384 × 384 otherwise 256 × 256.**

![Table 6. Results on MPII benchmark test set. ⊘ means pre-training on COCO dataset and fine-tuning on MPII dataset. ⊘↑ means training both on MPII and HSSK datasets.](image)
promising way to learn powerful and robust representation for human pose estimation and its downstream tasks.

4.3. Ablations

The importance of position embedding. Without position embedding, the 2D spatial structure information loses in Transformer. To explore its importance, we conduct experiments on TransPose-R-A3 models with three position embedding strategies: 2D sine position embedding, learnable position embedding, and w/o position embedding. As expected, the models with position embedding perform better, particularly for 2D sine position embedding, as shown in Tab. 4. But interestingly, TransPose w/o any position embedding only loses 1.3 AP, which suggests that 2D-structure becomes less important. See more details in supplementary.

Ablation study on the size of Transformer Encoder. | Model | #Layers | d | h | #Params | FLOPs | FPS | AP | AR |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TransPose-R</td>
<td>2</td>
<td>256</td>
<td>1024</td>
<td>4.4M</td>
<td>7.0G</td>
<td>174</td>
<td>69.6</td>
<td>75.0</td>
</tr>
<tr>
<td>TransPose-R</td>
<td>3</td>
<td>256</td>
<td>1024</td>
<td>4.2M</td>
<td>6.8G</td>
<td>141</td>
<td>71.7</td>
<td>77.1</td>
</tr>
<tr>
<td>TransPose-R</td>
<td>4</td>
<td>256</td>
<td>1024</td>
<td>4.0M</td>
<td>6.9G</td>
<td>138</td>
<td>72.6</td>
<td>78.0</td>
</tr>
<tr>
<td>TransPose-R</td>
<td>5</td>
<td>256</td>
<td>1024</td>
<td>3.8M</td>
<td>6.9G</td>
<td>126</td>
<td>72.2</td>
<td>77.6</td>
</tr>
<tr>
<td>TransPose-R</td>
<td>6</td>
<td>256</td>
<td>1024</td>
<td>3.6M</td>
<td>6.9G</td>
<td>118</td>
<td>71.7</td>
<td>77.7</td>
</tr>
</tbody>
</table>

4.4. Qualitative Analysis

The hyperparameter configurations for TransPose model might affect the its behavior in an unknown way. In this section, we choose trained models, types of predicted keypoints, depths of attention layers, and input images as controlled variables to observe the model behaviors.

The dependency preferences are different for models with different CNN extractors. To make comparisons between ResNet-S and HRNet-S based models, we use the trained models TP-R-A4 and TP-H-A4 performances as exemplars. Illustrated in Fig. 6, we choose two typical inputs A and B as examples and visualize the dependency areas defined in Sec. 3.3. We find that although the predictions from TP-R-A4 and TP-H-A4 are exactly the same locations of keypoints, TP-H-A4 can exploit multiple longer-range joints clues to predict keypoints. In contrast, TP-R-A4 prefers to attend to local image cues around the target joint. This characteristic can be further confirmed by the visualized affected areas in supplementary, in which keypoints have larger and non-local affected areas in TP-H-A4. Although such results are not as commonly expected, they reflect: 1) a pose estimator uses global information from long-range joints to localize a particular joint; 2) HRNet-S is better than ResNet-S at capturing long-range dependency relationships information (probably due to its multi-scale fusion scheme).

Dependences and influences vary for different types of keypoints. For keypoints in the head, localizing them mainly relies on visual clues from head, but TP-H-A4 also associates them with shoulders and the joints of arms. Notably, the dependencies of predicting wrists, elbows, knees or ankles have obvious differences for two models, in which TP-R-A4 depends on the local clues at the same side while TP-H-A4 exploits more clues from the joints on the symmetrical side. As shown in Fig. 6(b), Fig. 6(d), and Fig. 7, we can further observe that a pose estimator might gather strong clues from more parts to predict the target keypoint. This can explain why the model still can predict the location of an occluded keypoint accurately, and the occluded keypoint with ambiguity location will have less impact on the other predictions or larger uncertain area to rely on (e.g., the occluded left ankle – last map of Fig. 6(c) or Fig. 6(d)).
Attentions gradually focus on more fine-grained dependencies with the depth increasing. Observing all of attention layers (the 1,2,3-th rows of Fig. 7), we surprisingly find that even without the intermediate GT locations supervision, TP-H-A4 can still attend to the accurate locations of joints yet with more global cues in the early attention layers. For both models, with the depth increasing, the predictions gradually depend on more fine-grained image clues around local parts or keypoints positions (Fig. 7).

Image-specific dependencies and statistical commonalities for a single model. Different from the static relationships encoded in the weights of CNN after training, the attention maps are dynamic to inputs. As shown in Fig. 6(a) and Fig. 6(c), we can observe that despite the statistical commonalities on the dependency relationships for the predicted keypoints (similar behaviors for most common images), the fine-grained dependencies would slightly change according to the image context. With the existence of occlusion or invisibility in a given image such as input B (Fig. 6(c)), the model can still localize the position of the partially obscured keypoint by looking for more significant image clues and reduces reliance on the invisible keypoint to predict the other ones. It is likely that future works can exploit such attention patterns for parts-to-whole association and aggregating relevant features for 3D pose estimation or action recognition.

5. Conclusion

We explored a model – TransPose – by introducing Transformer for human pose estimation. The attention layers enable the model to capture global spatial dependencies efficiently and explicitly. And we show that such a heatmap-based localization achieved by Transformer makes our model share the idea with Activation Maximization. With lightweight architectures, TransPose matches state-of-the-art CNN-based counterparts on COCO and gains significant improvements on MPII when fine-tuned with small training costs. Furthermore, we validate the importance of position embedding. Our qualitative analysis reveals the model behaviors that are variable for layer depths, keypoints types, trained models and input images, which also gives us insights into how models handle special cases such as occlusion.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (61773117 and 62006041).


References


ising image classification models and saliency maps. In *ICLR (Workshop Poster)*, 2013. 2, 3, 4, 5


