Not All Tokens Are Equal: Human-centric Visual Analysis via Token Clustering Transformer

Wang Zeng[1], Sheng Jin[2,3], Wentao Liu[3], Chen Qian[3], Ping Luo[2], Wanli Ouyang[4], and Xiaogang Wang[1]

[1]The Chinese University of Hong Kong [2]The University of Hong Kong [3]SenseTime Research and Tetras.AI [4]The University of Sydney
{zengwang@link, xgwang@ee}.cuhk.edu.hk, {js20@connect, pluo@cs}.hku.hk,
{liuwentao, qianchen}@sensetime.com, wanli.ouyang@sydney.edu.au

Abstract

Vision transformers have achieved great successes in many computer vision tasks. Most methods generate vision tokens by splitting an image into a regular and fixed grid and treating each cell as a token. However, not all regions are equally important in human-centric vision tasks, e.g., the human body needs a fine representation with many tokens, while the image background can be modeled by a few tokens. To address this problem, we propose a novel Vision Transformer, called Token Clustering Transformer (TCFormer), which merges tokens by progressive clustering, where the tokens can be merged from different locations with flexible shapes and sizes. The tokens in TCFormer can not only focus on important areas but also adjust the token shapes to fit the semantic concept and adopt a fine resolution for regions containing critical details, which is beneficial to capturing detailed information. Extensive experiments show that TCFormer consistently outperforms its counterparts on different challenging human-centric tasks and datasets, including whole-body pose estimation on COCO-WholeBody and 3D human mesh reconstruction on 3DPW. Code is available at https://github.com/zengwang430521/TCFormer.git.

(a) Grid-based tokens in different stages
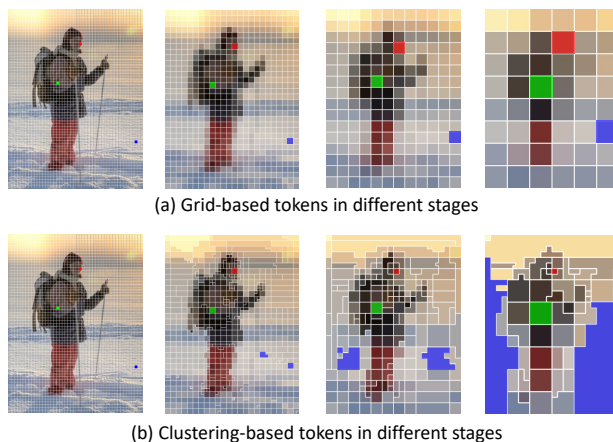

(b) Clustering-based tokens in different stages

Figure 1. Comparisons between vision tokens generated by (a) standard grids and (b) TCFormer. The token regions of different tokens, or the image regions represented by vision tokens, are visualized by different colors. From left to right, different images represent different stages. Grid-based tokens treat all regions equally as shown in (a). While the tokens in (b) treat image regions dynamically. Tokens distribute more densely on the human body. For background regions, a large area is represented by a single token (in blue), while for the regions containing important details, such as the face area, tokens with fine spatial sizes are used (in red).

1. Introduction

Human-centric tasks [12, 18, 21, 60, 69] of computer vision such as face alignment [2, 61, 76], human pose estimation [8, 22, 31, 37, 44, 45, 50, 54, 65, 67], and 3D human mesh reconstruction [27, 29, 52, 74] have drawn increasing research attention owing to their broad applications such as action recognition, virtual reality, and augmented reality.

Inspired by the success of transformers in natural language processing, vision transformers are recently developed to solve human-centric computer vision tasks and achieve state-of-the-art performance [33,35,40,66,71]. The properties of transformers such as the long-range attention between image patches are beneficial to model the relationship between different body parts and thus are critical in human-centric visual analysis.

Since the traditional transformers employed a sequence of tokens as input, most existing vision transformers follow this paradigm by dividing an input image into a regular and fixed grid, where each cell (image patch) is treated as a token as shown in Figure 1 (a). The grid-based token generation is simple and achieves great successes in many computer vision tasks [10, 38, 57] such as image recognition, object detection, and segmentation.

However, fixed grid based vision tokens are sub-optimal for human-centric visual analysis. In human-centric visual analysis, the image regions of the human body are more crucial than the image background, motivating us to represent different image regions by vision tokens with dynamic shape and size [1]. But the token regions of the grid-based vision tokens are rectangular areas with fixed location, shape and size. Uniform vision token distribution is not able to allocate more tokens to important areas.

To solve this problem, we propose a novel vision transformer, named Token Clustering Transformer (TCFormer), which generates tokens by progressive token clustering. TCFormer generates tokens dynamically at every stage. As shown in Figure 1 (b), it is able to generate tokens with various locations, sizes, and shapes. Firstly, unlike the grid-based tokens, tokens after clustering are not limited to the regular shape and can focus on important areas *e.g.*, the human body. Secondly, TCFormer dynamically generates tokens with appropriate sizes to represent different regions. For the regions full of important details such as the human face, tokens with finer size are allocated. In contrast, a single token (*e.g.*, the token in blue in Figure 1 (b)) is used to represent a large area of the background.

In TCFormer, every pixel in the feature map is initialized as a vision token at the first stage, whose token region is the region covered by the pixel. We progressively merge tokens with similar semantic meanings and obtain different numbers of tokens in different stages. To this end, we carefully design a Clustering Token Merge (CTM) block. Firstly, given tokens from the previous stage, CTM groups them by applying the k-nearest-neighbor based density peaks clustering algorithm [11] on the token features. Secondly, the tokens assigned to the same cluster are merged to a single token by averaging the token features. Finally, the tokens are fed into a transformer block for feature aggregation. The token region of the merged token is the union of the input token regions.

Aggregation of multi-stage features is proved to be beneficial for human-centric analysis [49, 71]. Most prior works [38, 57, 71] transform vision tokens to feature maps and aggregate features in the form of feature maps. However, when transforming our dynamic vision tokens to feature maps, multiple tokens may locate in the same pixel grid, causing the loss of details. To solve this problem, we propose a Multi-stage Token Aggregation (MTA) head, which is able to preserve image details in all stages in an efficient way. Specifically, the MTA head starts from the tokens in the last stage, and then progressively upsamples tokens and aggregates token features from the previous stage, until features in all the stages are aggregated. The aggregated tokens are in one-to-one correspondence with pixels in the feature maps and are reshaped to the feature maps for subsequent processing.

---

[1]We call the image region represented by a token as the token region and use token location, shape, and size to denote that of its token region.

We summarize our contributions as follows.

- We propose a Token Clustering Transformer (TCFormer), which generates vision tokens of various locations, sizes, and shapes for each image by progressive clustering and merging tokens. To the best of our knowledge, it is the first time that clustering is used for dynamic token generation.

- We propose a Multi-stage Token Aggregation (MTA) head to aggregate token features in multiple stages, reserving detailed information in all stages efficiently.

- Extensive experiments show that TCFormer consistently outperforms its counterparts on different challenging human-centric tasks and datasets, including whole-body pose estimation on COCO-WholeBody and 3D human mesh reconstruction on 3DPW.

2. Related Works

2.1. Transformers in Human-Centric Vision Tasks

Modeling the interaction between human and environment, and the relationships between body parts is a key point in human-centric vision tasks. With global receptive fields, transformers have achieved great success recently in 2d pose estimation [33, 40, 66, 71], 3d pose estimation [15, 19, 75] and 3d mesh reconstruction [34, 35]. Prior works can be divided into two categories, *i.e.* refining image features and aggregating part-wise features.

The first kind of method applies transformers to extract better image features. For example, TransPose [66] refines the image features extracted by CNN with extra transformer encoder layers. HRFormer [71] applies transformer blocks in HRNet [49] structure. Our proposed TCFormer also belongs to the first kind.

The second kind of method applies transformers to aggregate features for different parts. For example, TF-Pose [40] and TokenPose [33] use stacked transformers to generate keypoint-wise features, from which the keypoint location is predicted by regressing coordinates and heatmaps respectively. Mesh Graphormer [35] designs a head with transformers and graph convolution layers. The head aggregates joint-wise and vertex-wise features from image feature maps and predicts the 3D location of human joints and mesh vertices.

Most prior works in both categories generate vision tokens with fixed grid token region, which is sub-optimal for human-centric tasks. In contrast, the token regions of TCFormer are learned automatically rather than using a hand-crafted design. The learned token regions are flexible in location, shape and size according to the semantic meanings.
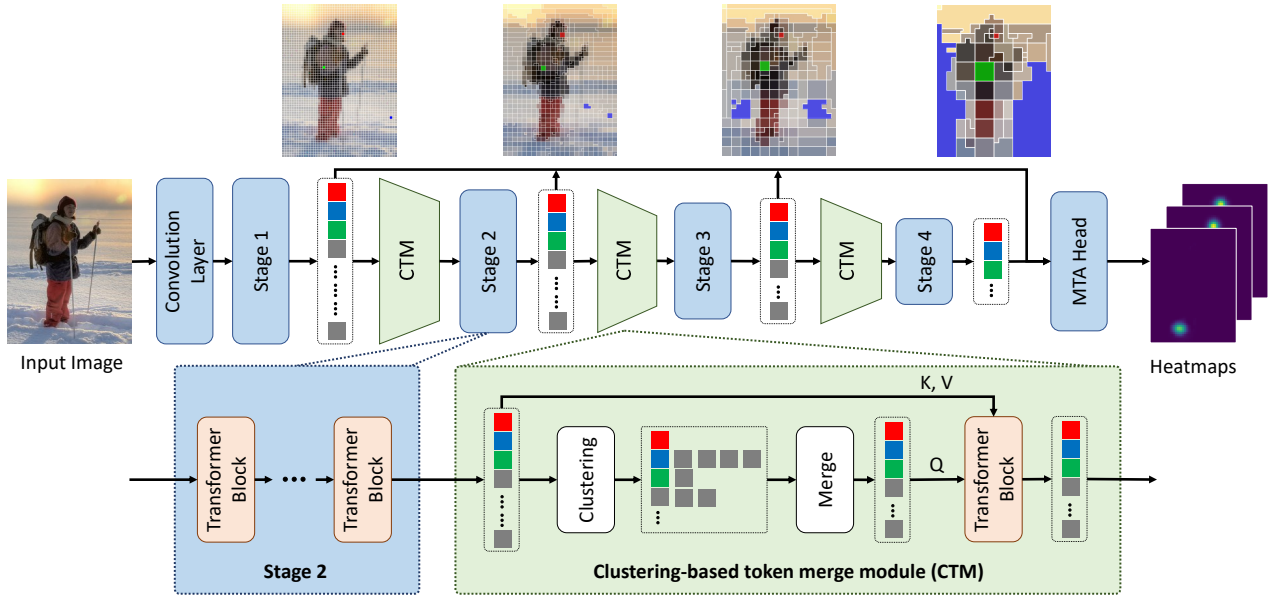
Figure 2. Overview of Token Clustering Transformer (TCFormer). Given an input image, feature map is extracted with a single convolution layer and pixels in the feature map is regarded as initial vision tokens. These initial vision tokens are processed in a multi-stage manner and every stage is composed of multiple stacked transformer blocks. At the end of every stage, vision tokens are merged with a Clustering-based Token Merge (CTM) block to generate tokens for the next stage. Finally, all vision tokens are fed into the Multi-stage Token Aggregation (MTA) head for predicting heatmaps. In the CTM block, input tokens are first clustered and then merged by weighted averaging. Merged tokens and original tokens are then fed into a transformer block for better feature aggregation.

2.2. Dynamic Token Generation

Recently, there are increasing exploration about dynamic token generation in vision transformers [47,56,58,72]. DynamicViT [47] and PnP-DETR [56] pick up important tokens by predicting token-level scores. For unimportant tokens, DynamicViT simply discards them, while PnP-DETR represents them with sparse feature vectors. DVT [58] builds vision tokens with grids in different resolutions according to the difficulty of classification. PS-ViT [72] builds tokens with a fixed grid size and progressively adjusts the grid centers during processing.

The methods mentioned above are all variants of grid-based token generation. They modify the number, resolution, and centers of grids specifically. In contrast, the token regions of our proposed TCFormer are not restricted by grid structure and are more flexible in three aspects, *i.e.* location, shape, and size. Firstly, we assign image regions to certain vision tokens based on the semantic similarity instead of the spatial proximity. Secondly, the image region of a token is not restricted to a rectangular shape. Regions with the same semantic meaning, even if they are non-adjacent, can be represented by a single token. Thirdly, vision tokens in the same stage may have different sizes. For example, in Figure 1 (b), the token for the background presents a large region, while the token for the human face only presents a small region. Such property is helpful for retaining important details among all the stages of TCFormer.

2.3. Clustering for Feature Aggregation

Clustering-based feature aggregation methods are wildly explored in research areas of point clouds [46] and graph representations [68]. PointNet++ [46] downsamples point clouds by farthest point sampling and then aggregates features of the k-nearest neighbors. DIFFPOOL [68] predicts soft cluster assignment matrices for an input graph, which are used to hierarchically coarsen the graph.

These methods are specially designed for point cloud and graph data, and cannot be directly applied to the image-based vision transformers. In contrast, we cluster the tokens in the different stages for image-based human-centric vision tasks. To the best of our knowledge, it is the first time that clustering is utilized for vision token generation.

3. Method

3.1. Overview Architecture

As shown in Figure 2, our proposed Token Clustering Transformer (TCFormer) follows the popular multi-stage architecture. TCFormer consists of 4 hierarchical stages and a Multi-stage Token Aggregation (MTA) head. Each stage contains several stacked transformer blocks. Between two adjacent stages, a Clustering-based Token Merge (CTM) block is inserted to merge tokens and generate tokens for the next stage. MTA head aggregates token features from all stages and outputs the final heatmaps.
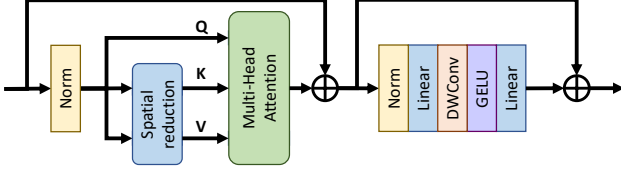
Figure 3. Structure of the transformer block in TCFormer. A spatial reduction module is added before the multi-head attention module to reduce the computational complexity. A depth-wise convolutional layer is added after the attention module to capture the local information.

3.2. Transformer Block

Figure 3 shows the transformer block used in TCFormer. Following [57], a spatial reduction layer is used to reduce the computational complexity. The spatial reduction layer first transforms the vision tokens to feature maps and then reduces the feature map resolution with a strided convolutional layer. The pixels in the processed feature maps, which are much fewer than the vision tokens, are fed into the multi-head attention module as keys and values. The multi-head attention module aggregates feature between tokens. Inspired by [32, 59, 71], we utilize a depth-wise convolutional layer to capture local feature and positional information, and remove explicit positional embedding.

3.3. Clustering-based Token Merge (CTM) Block

As shown in Figure 2, our Clustering-based Token Merge (CTM) block has two processes, *i.e. token clustering* and *feature merging*. We apply *token clustering* to group vision tokens into a certain number of clusters using token features, and then apply *feature merging* to merge the tokens in the same cluster into a single token.

Token Clustering. In the token clustering process, we utilize a variant of k-nearest neighbor based density peaks clustering algorithm (DPC-KNN) [11].

Given a set of tokens X, we compute the local density ρ of each token according to its k-nearest neighbors:

$$\rho_i = \exp\left(-\frac{1}{k}\sum_{x_j \in \mathrm{KNN}(x_i)} \|x_i - x_j\|_2^2,\right), \quad (1)$$

where $\mathrm{KNN}(x_i)$ denotes the k-nearest neighbors of a token i. x_i and x_j are their corresponding token features.

Then, for each token, we compute the distance indicator as the minimal distance between it and any other token with higher local density. For the token with the highest local density, its indicator is set as the maximal distance between it and any other tokens.

$$\delta_i = \begin{cases} \min_{j:\rho_j > \rho_i} \|x_i - x_j\|_2, & \text{if } \exists j \text{ s.t. } \rho_j > \rho_i \\ \max_j \|x_i - x_j\|_2, & \text{otherwise} \end{cases} \quad (2)$$

where δ_i is the distance indicator and ρ_i is the local density.

We combine the local density and the distance indicator to get the score of each token as $\rho_i \times \delta_i$. Higher scores mean higher potential to be cluster centers. We determine cluster centers by selecting the tokens with the highest scores, and then assign other tokens to the nearest cluster center according to the feature distances.

Feature Merging. For token features merging, an intuitive method is to average the token features in the cluster directly. However, even for the tokens with similar semantic meanings, the importance is not totally the same.

Inspired by [47], we introduce an importance score P to explicitly represent the importance of each token, which is estimated from the token features. The token features are averaged with the guidance of the token importance:

$$y_i = \frac{\sum_{j \in C_i} e^{p_j} x_j}{\sum_{j \in C_i} e^{p_j}}, \quad (3)$$

where C_i means the set of the i-th cluster, x_j and p_j are the original token features and the corresponding importance score respectively, and y_i is the features of the merged token. The token region of the merged token is the union of the original token regions.

As shown in Figure 2, the merged tokens are fed into a transformer block as queries Q, and the original tokens are used as keys K and values V. To ensure that important tokens contribute more to the output, the importance score P is added to the attention weight as follows:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + P\right)V, \quad (4)$$

where d_k is the channel number of the queries. We omit the multi-head setting and the spatial reduction layer here for clarity. Introducing the token importance score equips our CTM block with the capability to focus on the critical image features when merging vision tokens.

We adopt an efficient GPU implementation of CTM block. The clustering and feature merging parts only cost 9.4% of the forward time of TCFormer.

3.4. Multi-stage Token Aggregation (MTA) Head

Prior works [49, 71] prove the benefits of feature aggregation in multiple stages for human-centric vision tasks. In order to aggregate features, we propose a transformer-based Multi-stage Token Aggregation (MTA) head, which is able to maintain details in all the stages.

Figure 4 (a) shows the token upsampling process. During the token merging process (Section 3.3), each token is assigned to a cluster and each cluster is represented by a single merged token. We record the relationship between the original tokens and merged tokens. In the token upsampling process, we use the recorded information to copy the merged token features to the corresponding upsampled tokens. As shown in Figure 4 (b), after the token upsampling,

(a) Illustration of token upsampling process

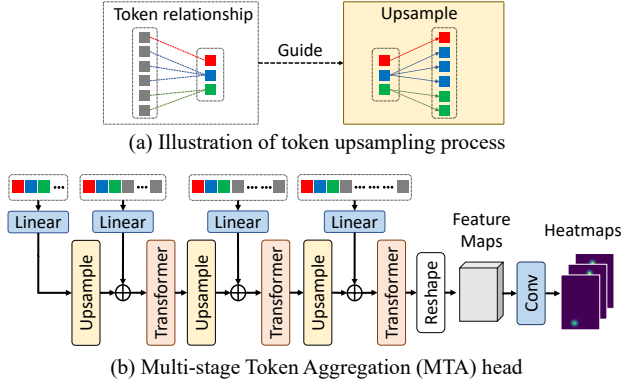(b) Multi-stage Token Aggregation (MTA) head

Figure 4. Multi-stage Token Aggregation (MTA) head. (a) In token upsampling process, we use the recorded token relationship to copy the merged token features to the corresponding upsampled tokens. (b) MTA head starts from the last stage and progressively aggregates features by stacked upsampling processes and transformer blocks. The final tokens are in one-to-one correspondence with the pixels in feature maps and reshaped to the feature maps for further processing.

MTA head adds the token features in the previous stage to the upsampled vision tokens. The vision tokens are then processed by a transformer block. Such processing is executed progressively until all vision tokens are aggregated. The final tokens, whose token region is a single pixel in the high-resolution feature map, can be easily reshaped to feature maps for further processing.

Why not CNN? Most prior works [38, 57, 71] transform vision tokens to feature maps first and aggregate multi-stage features with convolutional layers. However, as shown in Figure 5, vision tokens for the regions with important details, such as the human face, are in small size. Transforming vision tokens to low-resolution feature maps involves feature averaging of these tokens, which leads to the loss of details. Such loss can be avoided by transforming vision tokens in all stages to high-resolution feature maps, but it will lead to unacceptable complexity and memory cost. Our MTA head, which aggregates features in the form of tokens and transforms the final tokens to high-resolution feature maps at the end, preserves image details in all stages with relatively low complexity.

3.5. Implementation Details

Transform between Vision Tokens and Feature Maps. The convolutional process is non-trivial for irregular tokens. We need to transform tokens to feature maps before this process and perform the inverse transform after that. We regard every pixel in feature maps as a representation of a rectangular grid region. When transforming vision tokens to feature maps, we first find the related vision tokens for every pixel, *i.e.* the vision tokens whose token region overlaps with the grid region of the pixel. Then we average the token features according to the overlapped areas to get the image feature of every pixel. When transforming feature
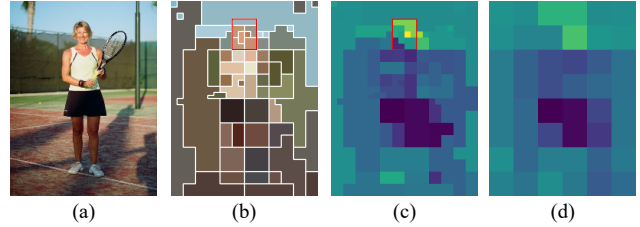

Figure 5. A typical example of vision tokens generated by TC-Former. (a) Input image. (b) Generated vision tokens in stage 4. Regions with the same color are represented by the same vision token. For the regions with details, such as the face area, vision tokens are with fine sizes. (c)(d)High resolution and low-resolution feature maps transformed from vision tokens. Transforming vision tokens to low-resolution feature maps causes detail loss.

maps to vision tokens, we directly set the token features as the average image features in the token region.

Cluster Numbers. In the CTM block, the cluster number can be any value smaller than the input token number. In this work, we set the cluster number to a quarter of the input token number to keep the token numbers in every stage the same as those of the prior works [38, 57].

4. Experiments

4.1. 2D Whole-body Pose Estimation

Whole-body pose estimation targets at localizing fine-grained keypoints on the entire human body including the face, the hands, and the feet, which requires the ability to capture detailed information.

Settings. We conduct experiments on COCO-WholeBody V1.0 dataset [23]. COCO-WholeBody dataset is built upon the popular COCO dataset [36] with additional whole-body pose annotations. The full pose contains 133 keypoints, including 17 for the body, 6 for the feet, 68 for the face, and 42 for the hands. Following [23, 36], we use OKS-based Average Precision (AP) and Average Recall (AR) for evaluation. We follow most of the default training and evaluation settings of mmpose [9] and replace Adam [28] with AdamW [39] with momentum of 0.9 and weight decay of 1×10^{-2}.

Results. Table 1 shows the experimental results comparing TCFormer with the state-of-the-art models on COCO-WholeBody V1.0 dataset [23]. The results show that the whole-body pose estimation accuracy of TCFormer (57.2% AP and 67.8% AR) is higher than those of the state-of-the-art top-down methods, *e.g.* HRNet [49], by a large margin.

The size of the hand in the input image is relatively small, which makes the estimation of hand keypoint extremely difficult and heavily reliant on the capability of the model in capturing details. As shown in Table 1, most models achieve much lower performance on the hand than the other parts. Our TCFormer achieves a large gain in hand keypoint estimation, *i.e.* 6.2% AP higher than HRNet-

Table 1. OKS-based Average Precision (AP) and Average Recall (AR) on the COCO-WholeBody V1.0 dataset. The baseline results are from MMPose [9]. '*' indicates multi-scale testing. ZoomNet[†] is trained with the COCO-WholeBody V0.5 training set.

Method	Resolution	body		foot		face		hand		whole-body	
		AP	AR	AP	AR	AP	AR	AP	AR	AP	AR
SN* [17]	480×480	0.427	0.583	0.099	0.369	0.649	0.697	0.408	0.580	0.327	0.456
OpenPose [4]	480×480	0.563	0.612	0.532	0.645	0.765	0.840	0.386	0.433	0.442	0.523
PAF* [5]	480×480	0.381	0.526	0.053	0.278	0.655	0.701	0.359	0.528	0.295	0.405
AE [43]+HRNet-w48 [49]	512×512	0.592	0.686	0.443	0.595	0.619	0.674	0.347	0.438	0.422	0.532
HigherHRNet-w48 [6]	512×512	0.630	0.706	0.440	0.573	0.730	0.777	0.389	0.477	0.487	0.574
ZoomNet[†] [23]	384×288	0.743	0.802	0.798	0.869	0.623	0.701	0.401	0.498	0.541	0.658
SBL-Res50 [63]	256×192	0.652	0.739	0.614	0.746	0.608	0.716	0.460	0.584	0.520	0.633
SBL-Res101 [63]	256×192	0.670	0.754	0.640	0.767	0.611	0.723	0.463	0.589	0.533	0.647
SBL-Res152 [63]	256×192	0.682	0.764	0.662	0.788	0.624	0.728	0.482	0.606	0.548	0.661
HRNet-w32 [49]	256×192	0.700	0.746	0.567	0.645	0.637	0.688	0.473	0.546	0.553	0.626
TCFormer w/o CTM	256×192	0.667	0.749	0.562	0.695	0.617	0.621	0.479	0.590	0.535	0.639
TCFormer w/o MTA Head	256×192	0.679	0.761	0.658	0.780	0.634	0.732	0.499	0.619	0.553	0.662
TCFormer (Ours)	256×192	0.691	0.770	0.698	0.813	0.649	0.746	0.535	0.650	**0.572**	**0.678**

Table 2. Mean Per Joint Position Error (MPJPE) and the error after Procrustes alignment (PA-MPJPE) on 3DPW and Human3.6M dataset. ↓ denotes lower is better. Our proposed method achieves results competitive with the state-of-the-art models. "*" means training without using 3DPW.

Method	3DPW		Human3.6M	
	PA-MPJPE↓	MPJPE↓	PA-MPJPE↓	MPJPE↓
HMR [27]*	76.7	130.0	56.8	88
SPIN [29]*	59.2	96.9	**41.1**	62.5
Zanfir *et al.* [73]	57.1	90.0	-	-
EFT [26]	52.2	-	43.8	-
DSR [13]	51.7	85.7	41.4	**62.0**
TCFormer (Ours)	**49.3**	**80.6**	42.8	62.9

random rotation, random flipping, and random color jittering. The model is trained with 8 GPUs with a batch size of 32 in each GPU for 80 epochs. We use Adam optimizer [28] with the learning rate of 2.5×10^{-4} and do not use learning rate decay. The settings are the same as that in [13, 26].

Results. We compare TCFormer with the state-of-the-art SMPL-based methods and show the results in Table 2. TC-Former outperforms most of the prior works [26, 27, 29, 73] with similar structures and model complexity. TCFormer even obtains competitive results compared with the DSR method that uses extra dense supervision [13]. The results of human mesh estimation further validate the effectiveness of TCFormer in capturing important image features. It works well not only on the dense prediction task, but also on the global feature based regression task.

w32 [49] and 5.3% AP higher than SBL-Res152 [63], which demonstrates the excellent capability of TCFormer in capturing critical image details with small sizes.

4.2. 3D Human Mesh Reconstruction

Prior works of 3D human mesh reconstruction can be divided into model-based [26, 27, 29, 29, 30, 73] and model-free methods [7, 34, 35, 42, 74]. We build a model-based method by combining TCFormer and HMR head [27].

Dataset. We evaluate our model on 3DPW dataset [53] and Human3.6M dataset [20]. 3DPW is composed of more than 51K frames with accurate 3D mesh in the wild, and Human3.6M is a large-scale indoor dataset for 3D human pose estimation. Mean Per Joint Position Error (MPJPE) and the error after Procrustes alignment (PA-MPJPE) are reported. Following the setting of [27], we train our model with a mixture of datasets, including Human3.6M [20], MPI-INF-3DHP [41], LSP [24], LSP-Extended [25], MPII [1] and COCO [36]. For fair comparisons with the recent methods [13, 26], we also use the training set of 3DPW and the pseudo-label provided by [26].

Settings. We crop the image with the ground truth bounding box and resize it to the resolution of 224×224. During training, we augment the data with random scaling,

4.3. 2D Face Keypoint Localization

Settings. We perform experiments on WFLW [61] dataset, which consists of $7,500$ training and $2,500$ testing images with 98 landmarks. The evaluation is conducted on the test set and several subsets: large pose, expression, illumination, make-up, occlusion, and blur. We use the normalized mean error (NME) as the evaluation metric and the inter-ocular distance for normalization. We apply the same training and evaluation settings as that of [55]. For fair comparisons, we use a lightweight version of TCFormer (TCFormer-Light) with similar model complexity as [55].

Results. As shown in Table 3, TCFormer achieves superior performance (4.28% NME), compared to the other state-of-the-art methods on the full test set and all the subsets. TCFormer even has lower error than the methods with extra information, such as PDB [14] which uses strong data augmentation and LAB [61] which uses extra boundary information. The performance on face alignment validates the versatility of TCFormer beyond human body estimation.

Table 3. NME on WFLW `Test` and 6 subsets: large pose (LP), expression (Expr.), illumination (Illu.), make-up (Mu.), occlusion (Occu.), and Blur. ↓ means lower is better. LAB [61] relies on extra boundary information (B). PDB [14] uses stronger data augmentation (DA).

| Method	Backbone	Test ↓	LP ↓	Expr. ↓	Illu. ↓	Mu. ↓	Occu. ↓	Blur ↓
ESR [3]	-	11.13	25.88	11.47	10.49	11.05	13.75	12.20
SDM [64]	-	10.29	24.10	11.45	9.32	9.38	13.03	11.28
CFSS [76]	-	9.07	21.36	10.09	8.30	8.74	11.76	9.96
DVLN [62]	VGG-16	6.08	11.54	6.78	5.73	5.98	7.33	6.88
HRNetV2 [55]	HRNetV2-W18	4.60	7.94	4.85	4.55	4.29	5.44	5.42
TCFormer (Ours)	TCFormer-Light	**4.28**	**7.27**	**4.56**	**4.18**	**4.27**	**5.18**	**4.87**
Model trained with `extra` info.								
LAB (w/ B) [61]	Hourglass	5.27	10.24	5.51	5.23	5.15	6.79	6.32
PDB (w/ DA) [14]	ResNet-50	5.11	8.75	5.36	4.93	5.41	6.37	5.81
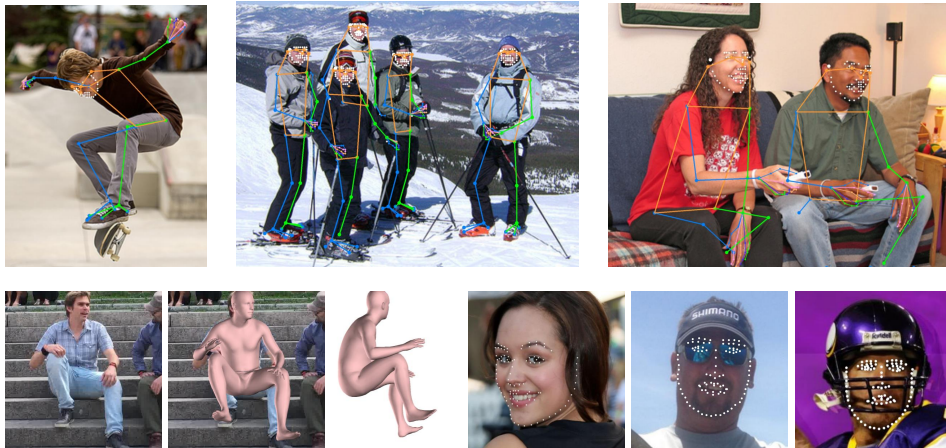
Figure 6. Example results of TCFormer on Whole-body pose estimation (top row), 3D mesh reconstruction (bottom left), and face alignment (bottom right). Overall, our method achieves satisfactory results on all tasks.

Table 4. Evaluation on ImageNet-1k `val`. All results are obtained using the input size of 224×224.

Method	#Param.	FLOPs	Top-1 Acc.
ResNet50 [16]	25.6M	4.1G	76.1
ResNet152 [16]	60.2M	11.6G	78.3
HRNet-W32 [55]	41.2M	8.3G	78.5
DeiT-Small/16 [51]	22.1M	4.6G	79.9
T2T-ViT$_t$-14 [70]	22.0M	6.1G	80.7
HRFormer-S [71]	13.5M	3.6G	81.2
Swin-T [38]	29.0M	4.5G	81.3
PVT-Large [57]	61.4M	9.8G	81.7
TCFormer (Ours)	25.6M	5.9G	**82.4**

4.4. Image Classification

In order to evaluate the versatility of TCFormer on general vision tasks, we also extend it to image classification.

Settings. We perform experiments on the ImageNet-1K dataset [48]. We apply the setting totally the same as [57]. TCFormer without the MTA head is trained from scratch for 300 epochs with batch size 128 and evaluated on the validation set with a center crop of 224×224 patch. For more details, please refer to the supplementary.

Results. Although the target of our model is not image classification, experimental results on ImageNet-1K show that TCFormer achieves competitive performance (82.4% Top-1 Acc.) compared with the state-of-the-art architectures, which indicates that our CTM block also works well in extracting general image features.

4.5. Ablation Study

We conduct ablative analysis on the task of whole-body pose estimation as shown in Table 1.

Effect of CTM. To validate the effect of CTM (Section 3.3), we build a baseline transformer network by replacing CTM with a strided convolutional layer, and find that the performance drops significantly (-3.7% AP and -3.9% AR). The performance drop for the parts relying on detailed information, such as the foot and the hand, is more significant than the others. The estimation AP for the foot and the hand decrease by 13.6% and 5.6% respectively, while the AP for the human body only drops 2.4%. It demonstrates that the performance drop is mainly caused by the loss of image details and validates the effectiveness of our CTM block in capturing image details in small size.

Effect of MTA Head. To validate the effect of MTA Head (Section 3.4), we replace it with a deconvolutional head [63] and notice a performance drop of -1.9% AP
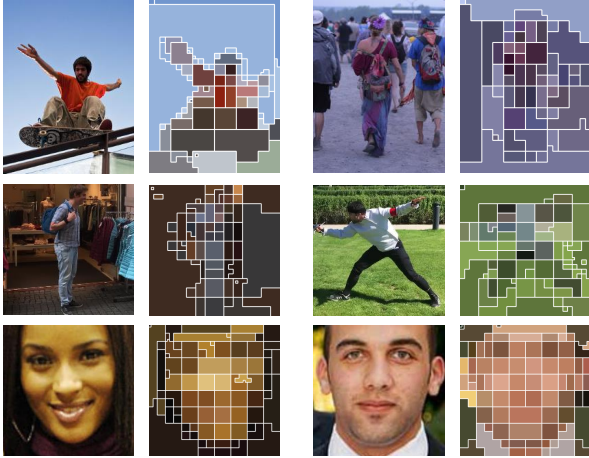
Figure 7. Vision tokens for whole-body pose estimation (the 1st row), 3D mesh reconstruction (the 2nd row), and face alignment (the 3rd row). For whole-body pose and 3D mesh estimation, TC-Former generates fine tokens for human face and hand. While for face alignment, fine tokens are used for face edges.

and -1.6% AR. Especially, the performance drop on foot (-4.0% AP) and hand (-3.6% AP) is more obvious than that of the body (-1.2% AP). The results show that MTA Head is beneficial to preserving image details of body parts.

4.6. Qualitative Evaluation

Figure 6 shows some qualitative results for human whole-body pose estimation, 3D mesh reconstruction, and face alignment. Figure 7 shows some examples of the vision token distribution in the above tasks. As shown in Figure 7, the vision tokens focus on the foreground regions and represent background areas with only very few tokens, even when the background is complex. The vision tokens with fine spatial size are used for the area with important details, for example, the face and hand regions in whole-body estimation and 3D mesh reconstruction. For face alignment, TCFormer allocates fine tokens to the face edge regions. The dynamic vision token allows TCFormer to capture image details more efficiently and achieve better performance.

5. Analysis

In this part, we provide an explanation of why TCFormer focuses on regions with important details. Take the face region in the human whole-body pose estimation task as an example, in order to distinguish the dense keypoints on the face, TCFormer tends to learn different features for different face areas. Since the features of the tokens representing different face parts are different, CTM block tends to group them into different clusters. So they are not merged with each other in the following merging process and are kept in a fine spatial size, which helps the feature learning in turn.

We verify the explanation by training two models with different tasks. The first one aims to estimate only the hand
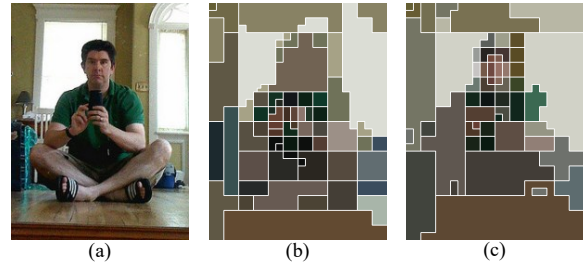

Figure 8. Tokens generated with different aims. We show (a) the input image and vision tokens generated by TCFormer that aims to estimate only (b) hand keypoints and (c) face keypoints. TC-Former adjusts the vision token distribution according to the task.

keypoints, while the other one aims to estimate only the face keypoints. We visualize the token distribution of these two models in Figure 8 (b) and (c), and find the token distribution to be task-specific, which matches our explanation.

6. Conclusions and Limitations

In this paper, we propose Token Clustering Transformer (TCFormer), a novel transformer-based architecture for human-centric vision tasks. We propose a Clustering-based Token Merge (CTM) block to equip the transformer with the capacity of preserving finer details in important regions and paying less attention to useless background information. Experiments show that our proposed method significantly improves upon its baseline model and achieves competitive performance on several human-centric vision tasks, *i.e.* whole-body pose estimation, human mesh recovery, and face alignment. The major limitation of TCFormer is that the computational complexity of KNN-DPC algorithm is quadratic with respect to the token number, which limits the speed of TCFormer with large input resolution. This problem can be mitigated by splitting tokens into multiple parts and performing part-wise token clustering. We envision that the proposed method is general and can be applied to a wide range of vision tasks, *e.g.* object detection, and semantic segmentation. Future works will focus on exploring the effectiveness of CTM on these vision tasks.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3686–3693, 2014. 6

[2] Xavier P Burgos-Artizzu, Pietro Perona, and Piotr Dollár. Robust face landmark estimation under occlusion. In *Int. Conf. Comput. Vis.*, pages 1513–1520, 2013. 1

[3] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. *Int. J. Comput. Vis.*, 107(2):177–190, 2014. 7

[4] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018. 6

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 6

[6] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5386–5395, 2020. 6

[7] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *Eur. Conf. Comput. Vis.*, pages 769–787, 2020. 6

[8] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1831–1840, 2017. 1

[9] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020. 5, 6

[10] 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. *Int. Conf. Learn. Represent.*, 2021. 1

[11] Mingjing Du, Shifei Ding, and Hongjie Jia. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowledge-Based Systems*, 99:135–145, 2016. 2, 4

[12] Haodong Duan, Kwan-Yee Lin, Sheng Jin, Wentao Liu, Chen Qian, and Wanli Ouyang. Trb: a novel triplet representation for understanding 2d human body. In *Int. Conf. Comput. Vis.*, pages 9479–9488, 2019. 1

[13] Sai Kumar Dwivedi, Nikos Athanasiou, Muhammed Kocabas, and Michael J Black. Learning to regress bodies from images using differentiable semantic rendering. In *Int. Conf. Comput. Vis.*, 2021. 6

[14] Zhen-Hua Feng, Josef Kittler, Muhammad Awais, Patrik Huber, and Xiao-Jun Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 6, 7

[15] Shreyas Hampali, Sayan Deb Sarkar, Mahdi Rad, and Vincent Lepetit. Handsformer: Keypoint transformer for monocular 3d pose estimation of hands and object in interaction. *arXiv preprint arXiv:2104.14639*, 2021. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 7

[17] Gines Hidalgo, Yaadhav Raaj, Haroon Idrees, Donglai Xiang, Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Single-network whole-body pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 6

[18] Kun Hu, Zhiyong Wang, Wei Wang, Kaylena A Ehgoetz Martens, Liang Wang, Tieniu Tan, Simon JG Lewis, and David Dagan Feng. Graph sequence recurrent neural network for vision-based freezing of gait detection. *IEEE Trans. Image Process.*, 29:1890–1901, 2019. 1

[19] Lin Huang, Jianchao Tan, Jingjing Meng, Ji Liu, and Junsong Yuan. Hot-net: Non-autoregressive transformer for 3d hand-object pose estimation. In *ACM Int. Conf. Multimedia*, pages 3136–3145, 2020. 2

[20] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7):1325–1339, 2013. 6

[21] Sheng Jin, Wentao Liu, Wanli Ouyang, and Chen Qian. Multi-person articulated tracking with spatial and temporal embeddings. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5664–5673, 2019. 1

[22] Sheng Jin, Wentao Liu, Enze Xie, Wenhai Wang, Chen Qian, Wanli Ouyang, and Ping Luo. Differentiable hierarchical graph grouping for multi-person pose estimation. In *Eur. Conf. Comput. Vis.*, pages 718–734. Springer, 2020. 1

[23] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *Eur. Conf. Comput. Vis.*, pages 196–214, 2020. 5, 6

[24] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Brit. Mach. Vis. Conf.*, 2010. doi:10.5244/C.24.12. 6

[25] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1465–1472, 2011. 6

[26] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. In *International Conference on 3D Vision (3DV)*, pages 42–52, 2021. 6

[27] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 1, 6

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Int. Conf. Learn. Represent.*, 2015. 5, 6

[29] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. *Int. Conf. Comput. Vis.*, 2019. 1, 6

[30] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4501–4510, 2019. 6

[31] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. In *Int. Conf. Comput. Vis.*, pages 11025–11034, 2021. 1

[32] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. 4

[33] Yanjie Li, Shoukui Zhang, Zhicheng Wang, Sen Yang, Wankou Yang, Shu-Tao Xia, and Erjin Zhou. Tokenpose: Learning keypoint tokens for human pose estimation. *Int. Conf. Comput. Vis.*, 2021. 1, 2

[34] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1954–1963, 2021. 2, 6

[35] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. *Int. Conf. Comput. Vis.*, 2021. 1, 2, 6

[36] 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 *Eur. Conf. Comput. Vis.*, 2014. 5, 6

[37] Wentao Liu, Jie Chen, Cheng Li, Chen Qian, Xiao Chu, and Xiaolin Hu. A cascaded inception of inception network with attention modulated feature fusion for human pose estimation. In *AAAI*, 2018. 1

[38] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *Int. Conf. Comput. Vis.*, 2021. 1, 2, 5, 7

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *Int. Conf. Learn. Represent.*, 2019. 5

[40] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, and Zhibin Wang. Tfpose: Direct human pose estimation with transformers. *arXiv preprint arXiv:2103.15320*, 2021. 1, 2

[41] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *International Conference on 3D vision (3DV)*, pages 506–516, 2017. 6

[42] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *Eur. Conf. Comput. Vis.*, pages 752–768, 2020. 6

[43] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Adv. Neural Inform. Process. Syst.*, 2017. 6

[44] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Eur. Conf. Comput. Vis.*, 2016. 1

[45] Wanli Ouyang, Xiao Chu, and Xiaogang Wang. Multi-source deep learning for human pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2329–2336, 2014. 1

[46] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inform. Process. Syst.*, 2017. 3

[47] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Adv. Neural Inform. Process. Syst.*, 2021. 3, 4

[48] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. 7

[49] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5693–5703, 2019. 2, 4, 5, 6

[50] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1653–1660, 2014. 1

[51] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *ICML*, 2021. 7

[52] Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *Eur. Conf. Comput. Vis.*, 2018. 1

[53] Timo von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Eur. Conf. Comput. Vis.*, pages 601–617, 2018. 6

[54] Jiahang Wang, Sheng Jin, Wentao Liu, Weizhong Liu, Chen Qian, and Ping Luo. When human pose estimation meets robustness: Adversarial algorithms and benchmarks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11855–11864, 2021. 1

[55] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 6, 7

[56] Tao Wang, Li Yuan, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Pnp-detr: Towards efficient visual analysis with transformers. In *Int. Conf. Comput. Vis.*, pages 4661–4670, 2021. 3

[57] 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. *Int. Conf. Comput. Vis.*, 2021. 1, 2, 4, 5, 7

[58] Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic vision transformers with adaptive sequence length. *Adv. Neural Inform. Process. Syst.*, 2021. 3

[59] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *Int. Conf. Comput. Vis.*, 2021. 4

[60] Size Wu, Sheng Jin, Wentao Liu, Lei Bai, Chen Qian, Dong Liu, and Wanli Ouyang. Graph-based 3d multi-person pose estimation using multi-view images. In *Int. Conf. Comput. Vis.*, pages 11148–11157, 2021. 1

[61] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face

alignment algorithm. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 1, 6, 7

[62] Wenyan Wu and Shuo Yang. Leveraging intra and inter-dataset variations for robust face alignment. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 150–159, 2017. 7

[63] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Eur. Conf. Comput. Vis.*, 2018. 6, 7

[64] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 532–539, 2013. 7

[65] Lumin Xu, Yingda Guan, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. Vipnas: Efficient video pose estimation via neural architecture search. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16072–16081, 2021. 1

[66] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Towards explainable human pose estimation by transformer. *Int. Conf. Comput. Vis.*, 2021. 1, 2

[67] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *Int. Conf. Comput. Vis.*, pages 1281–1290, 2017. 1

[68] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *Adv. Neural Inform. Process. Syst.*, 2018. 3

[69] Kaimin Yu, Zhiyong Wang, Markus Hagenbuchner, and David Dagan Feng. Spectral embedding based facial expression recognition with multiple features. *Neurocomputing*, 129:136–145, 2014. 1

[70] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *Int. Conf. Comput. Vis.*, 2021. 7

[71] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution transformer for dense prediction. *Adv. Neural Inform. Process. Syst.*, 2021. 1, 2, 4, 5, 7

[72] Xiaoyu Yue, Shuyang Sun, Zhanghui Kuang, Meng Wei, Philip HS Torr, Wayne Zhang, and Dahua Lin. Vision transformer with progressive sampling. In *Int. Conf. Comput. Vis.*, pages 387–396, 2021. 3

[73] Andrei Zanfir, Eduard Gabriel Bazavan, Hongyi Xu, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Weakly supervised 3d human pose and shape reconstruction with normalizing flows. In *Eur. Conf. Comput. Vis.*, pages 465–481, 2020. 6

[74] Wang Zeng, Wanli Ouyang, Ping Luo, Wentao Liu, and Xiaogang Wang. 3d human mesh regression with dense correspondence. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7054–7063, 2020. 1, 6

[75] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. *Int. Conf. Comput. Vis.*, 2021. 2

[76] Shizhan Zhu, Cheng Li, Chen Change Loy, and Xiaoou Tang. Face alignment by coarse-to-fine shape searching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4998–5006, 2015. 1, 7