Dynamic Support Information Mining for Category-Agnostic Pose Estimation

Pengfei Ren, Yuanyuan Gao, Haifeng Sun, Qi Qi, Jingyu Wang[*], Jianxin Liao[*]
State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications
{rpf, gaoyuanyuan, hfsun, qiqi8266, wangjingyu, liaojx}@bupt.edu.cn

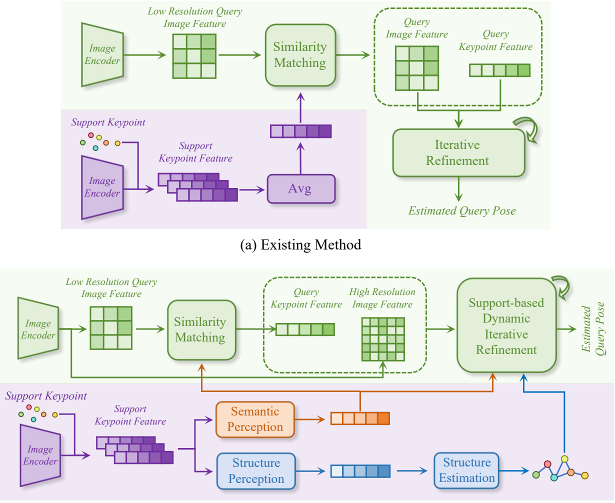
Abstract

Category-agnostic pose estimation (CAPE) aims to predict the pose of a query image based on few support images with pose annotations. Existing methods achieve the localization of arbitrary keypoints through similarity matching between support keypoint features and query image features. However, these methods primarily focus on mining information from the query images, neglecting the fact that support samples with keypoint annotations contain rich category-specific fine-grained semantic information and prior structural information. In this paper, we propose a Support-based Dynamic Perception Network (SDPNet) for the robust and accurate CAPE. On the one hand, SDPNet models complex dependencies between support keypoints, constructing category-specific prior structure to guide the interaction of query keypoints. On the other hand, SDPNet extracts fine-grained semantic information from support samples, dynamically modulating the refinement process of query. Our method outperforms existing methods on MP-100 dataset by a large margin.

1. Introduction

Pose estimation is a fundamental task in computer vision, aimed at estimating the coordinates of predefined semantic parts, such as human bodies [22, 26, 42, 44], hands [41, 54], faces [3, 43], animals [19, 47], and vehicles [35]. It is crucial for human-computer interaction, human behavior analysis, autonomous driving, and biomedicine. However, most current pose estimation methods are category-specific and fail to estimate poses of categories not seen during training. For new categories, these methods require collecting large amounts of annotated data for network training, or even adopting different network architectures, which is time-consuming and resource-intensive.

Recently, Category-Agnostic Pose Estimation (CAPE) [45] has been proposed to achieve universal pose estimation with a single model. Specifically, for an unseen object, given few support images with pose annotations, CAPE aims to predict keypoint locations in query images without re-training. As shown in Fig. 1, existing methods [33, 45] estimate the initial query keypoint positions through the similarity matching between keypoint features and query image features. Then, they iteratively refine the query keypoint positions through self-interaction among query keypoints and cross-interaction between query keypoint features and query image features. Although existing methods have made significant progress in CAPE, they still struggle to handle complex occlusions, similar appearances, and a wide range of pose variations.



(a) Existing Method

(b) Our Method

Figure 1. Comparison with existing methods. Previous methods mainly focus on information mining of query samples. Our method fully explores the support information, allowing support features to deeply participate in the iterative refinement.

The support samples serve as an explicit clue for pose estimation of novel categories, containing fine-grained visual information and category-related prior structural information. However, existing methods primarily use support in-

[*]Corresponding author

formation to initialize query keypoint features, significantly neglecting the potential of support samples. On the one hand, the similarity matching process is easily disrupted by complex occlusions or similar appearances, which is difficult to be refined through the interaction between query image features and query keypoint features. On the other hand, aggregating multiple support keypoint features through averaging ignores the complex inter- and intra-sample dependencies, making the refinement process susceptible to interference from low-quality keypoint features.

To address these problems, we propose a Support-based Dynamic Perception Network (SDPNet) for the robust and accurate CAPE. SDPNet extensively exploits the category-specific semantic information and structural information in support samples, to guide the similarity matching and iterative refinement of the query sample. On the one hand, SDPNet models the complex dependencies between keypoints through cross-sample interaction, and generates a category-specific prior structure. Based on the estimated prior structures, SDPNet adopts a Graph Convolution Network (GCN) to perform directional information passing between query keypoints, significantly enhancing the robustness of query keypoint features against occlusions and similarities. On the other hand, SDPNet constructs semantic-aware keypoint features with category consensus through cross-sample interaction. Based on the semantic-aware keypoint features, SDPNet dynamically modulates the interaction process between query keypoint features and query visual feature maps, activates category-relevant visual features and suppresses irrelevant features.

We evaluate our method on MP-100 dataset [45],which is currently the only public dataset for the CAPE task. Our method outperforms state-of-the-art (SOTA) methods (POMNet [33] and CapeFormer [45]) by a large margin. Our contributions can be summarized as follows:

- For the CAPE task, our method is the first to focus on mining and utilizing the semantic and structural information of support samples.
- Our method dynamically predicts category-related structures using support keypoint features, subsequently guiding the self-interaction among query keypoint features.
- Our method leverages semantic-aware support keypoint features to dynamically modulate the interaction between query keypoint features and query image features.

2. Related Works

2.1. Category-specific Pose Estimation

Pose estimation is a fundamental task in computer vision, aimed at locating the predefined semantic parts of an object. For a long time, pose estimation methods have focused on specific categories or super-categories, such as human bodies [22, 26, 42, 44], faces [3, 43], hands [41, 54], and animal poses [19, 47]. These methods can be divided into heatmap-based approaches [5, 8, 17, 26, 36, 38, 44, 48] and regression-based approaches [21, 28, 54]. Regression-based methods use an encoder structure to directly regress keypoint coordinates from images, while heatmap-based methods adopt an encoder-decoder structure to predict a dense pose representation, such as likelihood maps. These two types of methods mainly focus on building more powerful backbone [8, 26], designing better loss functions [17, 21], and adopting better pose representations [5, 38, 48]. However, these methods require a large amount of training data and lack the ability to detect keypoints of novel objects.

2.2. Category-Agnostic Pose Estimation

Generic visual models adopt image inpainting [1, 40], image editing [15], or the generation of serialized tokens [6, 16, 24, 39] as agent target to achieve multiple visual tasks, including pose estimation, object detection, semantic segmentation, depth estimation, and image generation. Although these methods are also category-agnostic, they primarily focus on how to perform various vision tasks consistently, overlooking the uniqueness of the pose estimation task. POMNet [45] is the first method designed specifically for category-agnostic pose estimation tasks. Given few support samples, POMNet achieves the query keypoints localization through a similarity matching process between keypoint features and query image features. The similarity matching process is susceptible to interference from similar appearances and occlusions. Thus, CapeFormer [33] proposes an attention-based iterative refinement process, leveraging the query image feature to progressively enhance the query keypoint features, significantly improving the accuracy of pose estimation. However, both methods substantially neglect the rich semantic information and prior structural information in the support samples.

2.3. Graph Convolution Networks

Graph Convolution Networks (GCN) have significant applications in multiple tasks, including action recognition [9, 32, 46], 3D reconstruction [10, 30, 52], human pose estimation [7, 23, 49, 50, 56] and hand pose estimation [4, 11, 12, 14, 20]. These works mainly use GCN to perform information passing between joints. For example, Cai *et al.* [4] adopt a hierarchical GCN to estimate 3D pose from a short sequence of 2D poses; Ren *et al.* [29] use GCN for cross-view hand joint feature interaction. Most of these methods depend on the physical structure between joints, which limits their flexibility. To solve this problem, some works [31, 51, 55] propose using data-driven methods to learn a adjacency matrix or dynamically predict the adjacency matrix based on joint features. However, these methods can still only handle a single category. Our method adopts a mask reconstruction task that dynamically predicts

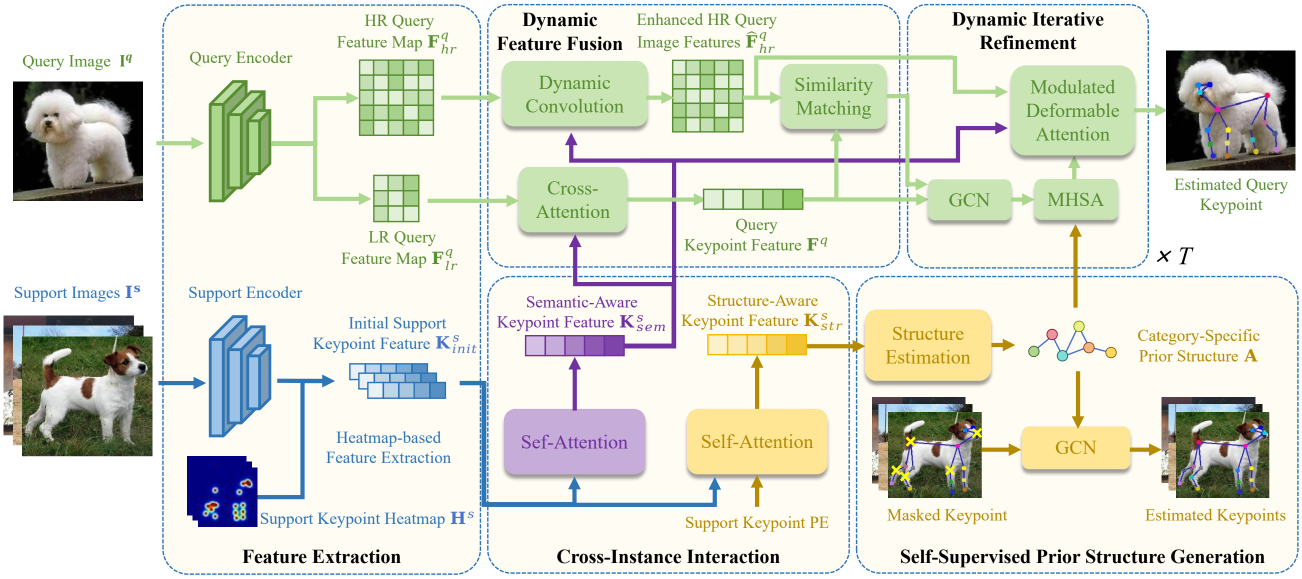

Query Image $\mathbf{I}^q$
Query Encoder
HR Query Feature Map $\mathbf{F}^q_{hr}$
Dynamic Feature Fusion
Enhanced HR Query Image Features $\widehat{\mathbf{F}}^q_{hr}$
Dynamic Convolution
Similarity Matching
Dynamic Iterative Refinement
Modulated Deformable Attention
Estimated Query Keypoint
Cross-Attention
Query Keypoint Feature $\mathbf{F}^q$
GCN
MHSA
LR Query Feature Map $\mathbf{F}^q_{lr}$
$\times T$
Support Images $\mathbf{I}^s$
Support Encoder
Initial Support Keypoint Feature $\mathbf{K}^s_{init}$
Semantic-Aware Keypoint Feature $\mathbf{K}^s_{sem}$
Structure-Aware Keypoint Feature $\mathbf{K}^s_{str}$
Structure Estimation
Category-Specific Prior Structure $\mathbf{A}$
Heatmap-based Feature Extraction
Sef-Attention
Self-Attention
Masked Keypoint
GCN
Estimated Keypoints
Support Keypoint Heatmap $\mathbf{H}^s$
Support Keypoint PE
Feature Extraction
Cross-Instance Interaction
Self-Supervised Prior Structure Generation

Figure 2. Overall Framework. Support information deeply participates in the dynamic feature fusion process and the dynamic iterative refinement process. 'HR' and 'LR' represent high resolution and low resolution, while 'PE' stands for positional encoding.

category-related structures based on support features.

3. Method

Category-Agnostic Pose Estimation (CAPE) [45] aims to estimate the pose of a query image based on few support images. Specifically, we denote the query image as $\mathbf{I}^q \in \mathbb{R}^{H\times W\times 3}$, the support images as $\mathbf{I}^s \in \mathbb{R}^{H\times W\times 3}$, and the number of support samples as N. Support samples provide rich clues for the pose estimation of the query sample. On the one hand, support images contain category-related visual information, such as color, texture and shape. On the other hand, the support keypoint coordinates contain category-related prior structure information, such as kinematic dependencies and symmetry relationships. Therefore, we propose a Support-based Dynamic Perception Network (SDPNet) to mine the semantic and structural information of support samples.

Similar to CapeFormer [33], SDPNet adopts a similarity matching to generate initial query keypoint coordinates and adopts an iterative refinement process to refine the query keypoint coordinates progressively. Differently, in SDPNet, the information of support samples is deeply integrated into the similarity matching and the iterative refinement. Firstly, SDPNet utilizes a cross-instance interaction module to extract semantic-aware and structure-aware keypoint features. Then, SDPNet adopts an information fusion module and a dynamic convolution module to construct query keypoint features and high-resolution query image features, respectively. Finally, under the guidance of two types of support keypoint features, SDPNet dynamically enhances query keypoint features and iteratively refines the query pose. T represents the iteration numbers.

3.1. Keypoint Feature Construction

SDPNet adopts a shared backbone to extract visual feature maps from the support images and the query image respectively. As shown in Fig. 2, the query image $\mathbf{I}^q$ is fed to the backbone to extract the high-resolution query feature map $\mathbf{F}^q_{hr} \in \mathbb{R}^{H/4\times W/4\times C}$ and low-resolution query feature map $\mathbf{F}^q_{lr} \in \mathbb{R}^{H/32\times W/32\times C}$, where the C represents the number of channels. The support image $\mathbf{I}^s$ is fed to the backbone to extract the support feature map $\mathbf{F}^s \in \mathbb{R}^{H/4\times W/4\times C}$. Then, we extract initial support keypoint features $\mathbf{K}^s_{init} \in \mathbb{R}^{K\times C}$ from the support feature maps $\mathbf{F}^s$ using keypoint heatmaps $\mathbf{H}^s \in \mathbb{R}^{H\times W\times K}$ as previous work [33, 45], where the K represents the number of keypoints.

We construct semantic-aware keypoint features $\mathbf{K}^s_{sem} \in \mathbb{R}^{K\times C}$ and structure-aware keypoint features $\mathbf{K}^s_{str} \in \mathbb{R}^{K\times C}$ based on initial support keypoint features $\mathbf{K}^s_{init}$. It is worth mentioning that the same semantic parts of different instances within the same category may vary significantly in shape, color, texture, and other attributes. Therefore, using average operations to aggregate keypoint information is unreliable. We adopt the self-attention mechanism [37] to perform cross-instance information interaction, thereby obtaining the support keypoint features with category consensus. As shown in Fig. 3, we first perform intra-instance keypoint interactions, followed by inter-instance keypoint interactions. Notably, during inter-instance interactions, we introduce instance order encoding and keypoint identifier encoding [33] to construct the query and key, thereby dis-

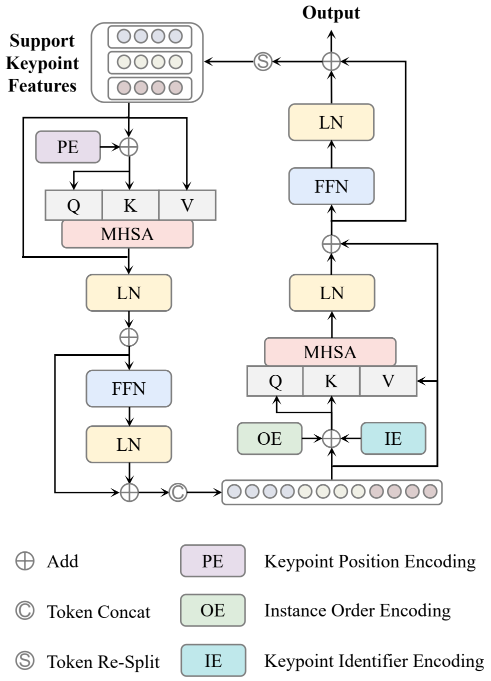

Output
Support Keypoint Features
PE
Q
K
V
MHSA
LN
FFN
LN
LN
MHSA
Q
K
V
OE
IE
FFN
LN
⊕ Add
© Token Concat
Ⓢ Token Re-Split
PE Keypoint Position Encoding
OE Instance Order Encoding
IE Keypoint Identifier Encoding

Figure 3. Illustration of the cross-instance interaction. We iteratively perform intra-instance keypoint interactions and then perform inter-instance keypoint interactions. Here, we use the 3-shot setting for illustration.

tinguishing between different instances and different keypoints. For structure-aware keypoint features $\mathbf{K}^s_{str}$, we enhance the initial keypoint features $\mathbf{K}^s_{init}$ with keypoint positional encoding and adopt another self-attention module for cross-instance information interaction.

The quality of the support keypoint features from different instances varies due to occlusion. Therefore, we utilize an instance-wise attention mechanism to obtain the aggregated semantic-aware keypoint features $\mathbf{K}^s_{sem}$ and structure-aware keypoint features $\mathbf{K}^s_{str}$.

3.2. Semantic-Aware Feature Fusion

Since different instances in the support images and the query image may have considerable differences in color, texture, and pose, it is difficult to directly match the support keypoint features and the query image features. Cape-Former [33] adopts a self-attention module to fuse support keypoint features and query image features, thereby reducing the gap in embedding space.

Since the computational complexity of the self-attention mechanism will grow quadratically with the length of the processing sequence, existing methods can only perform feature interaction with low-resolution feature maps. However, low-resolution feature maps need more fine-grained visual information, making it challenging to achieve high-precision pose estimation. To solve this problem, we use both low-resolution query feature maps and high-resolution query feature maps. Specifically, similar to [33], we adopt a self-attention module to perform the fusion of support keypoint features and low-resolution query feature maps to construct query keypoint features. At the same time, we feed the high-resolution query feature maps to the subsequent iterative refinement process to provide fine-grained visual clues for enhancing query keypoint features and refining query poses. To efficiently activate category-related visual features and suppress irrelevant features, we perform dynamic feature embedding based on support sample information. Specifically, we generate category-specific convolution kernels based on semantic-aware keypoint features $\mathbf{K}^s_{sem}$. Following [2, 27], we set convolution kernel size S to 7 and we adopt the depth-wise convolution kernels $\mathbf{D} \in \mathbb{R}^{C \times S \times S}$ to reduce computational complexity:

$$\mathbf{D} = FC(GAP(\mathbf{K}^s_{sem})), \tag{1}$$

where GAP denotes global average pooling and FC is a fully connected layer. With the predicted dynamic convolution kernels $\mathbf{D}$, enhanced high-resolution query feature map $\hat{\mathbf{F}}^q_{hr}$ is calculated as follow:

$$\hat{\mathbf{F}}^q_{hr} = ReLU(BN(\mathbf{D} *_d \mathbf{F}^q_{hr})), \tag{2}$$

where $*_d$ is the depth-wise convolution. $ReLU$ and BN represent ReLU function and batch normalization [18].

3.3. Dynamic Iterative Refinement

The similarity matching process focuses on the similarity of local visual features, so it is susceptible to interference from similar appearance and occlusion. For example, symmetrically similar parts, such as eyes, ears, and limbs, are prone to confusion in similarity matching. Therefore, we adopt a dynamic, iterative refinement process to enhance the query keypoint features and refine the query keypoint positions. Specifically, we utilize dynamic GCN and Multi-Head Self-Attention (MHSA) for query keypoint self-interaction. In addition, we utilize a modulated deformable attention mechanism to extract fine-grained visual information from high-resolution feature maps $\hat{\mathbf{F}}^q_{hr}$, and then update the query keypoint features.

Dynamic Graph Convolution. Each category has its unique prior structure, including physical connections, such as kinematic dependencies between key points, and spatial relationships, such as symmetry. Performing keypoint information interaction based on the prior structure can alleviate the interference of local occlusion and absences. Therefore, we utilize GCN to perform keypoint information passing based on category-specific prior structures and then adopt the MHSA to perform non-local keypoint interaction. Specifically, we dynamically generate category-specific prior structures $\mathbf{A} \in \mathbb{R}^{K \times K}$ based on structure-aware support keypoint features $\mathbf{K}^s_{str}$ as follow:

$$\mathbf{A} = \sum_i Softmax((\mathbf{W}_{\theta_i}\mathbf{K}^s_{str})^T\mathbf{W}_{\varphi_i}\mathbf{K}^s_{str}), \quad (3)$$

where $\mathbf{W}_{\theta_i}$ and $\mathbf{W}_{\varphi_i}$ are two learnable matrices used for feature embedding; i represents different embedding spaces. As shown in Fig. 2, we adopt a mask reconstruction task for self-supervised training. Specifically, we randomly mask out some support keypoints, and use a GCN to reconstruct keypoints. Meanwhile, we adopt a sparse constraint on the predicted structure to avoid redundant edges.

As mentioned in previous work [23], in order to enhance the modeling ability of graph convolution, different feature transformation matrices need to be used for different keypoints before feature aggregation. To improve the modeling capabilities without introducing too many parameters, Zou *et al*. [55] proposed to use a learnable modulation matrix $\mathbf{M} \in \mathbb{R}^{K\times C}$ to achieve disentangled transformation of the keypoint feature. Given a query keypoint feature $\mathbf{K}^q_i$, the graph convolution operation is defined as follow:

$$\mathbf{K}^q_i = ReLU(\sum_j(\mathbf{M}_j \odot \mathbf{W})\mathbf{K}^q_j\mathbf{A}_{ij}), \quad (4)$$

where $\mathbf{W}$ is a shared feature transformation matrix and $\odot$ represents Hadamard product. The value of $\mathbf{M}$ is fixed once training is completed. In the CAPE task, the semantics of each position in the keypoint features are not fixed. To solve this problem, we dynamically generate modulation matrices based on query keypoint features to achieve efficient disentangled transformation of keypoint features as follows:

$$\mathbf{M}_j = Sigmoid(MLP(\mathbf{K}^q_j)), \quad (5)$$

where MLP stands for a multi-layer perception consisting of multiple FC layers and ReLU activation layers. Specifically, within a single refinement stage, this modulation matrix is shared by all graph convolution layers.

Modulated Deformable Attention. In order to reduce the computational complexity, we utilize the deformable attention [53] for sparse image feature sampling and keypoint feature update. Each keypoint predicts a set of offset vectors based on its own information, then performs sparse image feature sampling based on these offset vectors to update its features. However, we observe that some keypoints exhibit excessive confidence, sampling features within a small range around themselves, which may lead to local optima. To address this, we utilize support keypoint information to modulate the sampling range, encouraging low-quality keypoints to access a broader range. As shown in Fig. 4, for the support and query features of each keypoint, we compare the differences in their embedded features and predict a modulation coefficient matrix $\mathbf{C} \in \mathbb{R}^{K\times 1}$ based on the difference as follow:

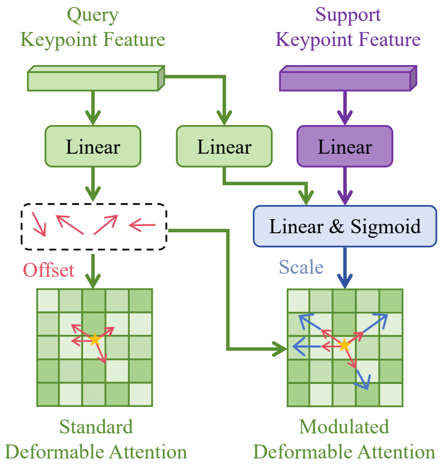


Figure 4. Illustration of the modulated deformable attention. We focus on explaining the differences in feature sampling between standard and modulated deformable attention, hence omitting the process of weighted aggregation of sampled features. For simplicity, we only draw a single keypoint feature.

$$\mathbf{C} = 2 * Sigmoid(MLP(\mathbf{W}_{\phi_0}\mathbf{K}^q_{str} - \mathbf{W}_{\phi_1}\mathbf{K}^s_{sem})), \quad (6)$$

where $\mathbf{W}_{\phi_i}$ are learnable matrices used for feature embedding. $Sigmoid$ function transforms the modulation coefficients to a range between 0 and 1.

Query Pose Refinement. We use the refined query keypoint features to predict the keypoint coordinates. Similar to CapeFormer [33], we predict the offset of keypoints during the iterative refinement. We use the output of the last refinement stage as the final prediction result.

3.4. Loss Function

We supervise three network parts, including similarity matching, iterative refinement, and prior structure generation. For similarity matching and iterative refinement, we supervise the similarity heatmap and estimated query keypoint coordinates, respectively. The supervision loss function is the same as CapeFormer [33]. For the prior structure generation, we mainly adopt support pose reconstruction loss and structure sparsity loss. For N support instances, we define the support pose reconstruction loss L_{pose} as:

$$L_{pose} = \frac{1}{N}\frac{1}{K}\sum_{n=1}^{N}\sum_{k=1}^{K}\left|\mathbf{P}^s_{nk} - \widetilde{\mathbf{P}}^s_{nk}\right|_1, \quad (7)$$

Given the estimated category-specific prior structure matrix, the structure sparsity loss L_{str} is defined as:

$$L_{str} = \frac{1}{K^2}\sum_{i=1}^{K}\sum_{j=1}^{L}\left|\mathbf{A}_{ij}\right|_1. \quad (8)$$

4. Experiments

4.1. Dataset and Metric

We train and evaluate our method on the MP-100 dataset [33], which is currently the only public dataset for CAPE tasks. MP-100 contains 100 sub-categories and 8 super-categories, with a total of 18K images and 20K annotations. The number of annotated keypoints covers a wide range, from 8 to 68. Following previous methods [33, 45], the dataset is divided into 5 splits. In each split, there is no overlap of categories between the training, validation, and test sets. We use the Probability of Correct Keypoint (PCK) as the evaluation metric. Following previous methods [33, 45], we use PCK under the 0.2 threshold as default.

4.2. Implementation Details

We train and evaluate our method on a computer with an AMD Ryzen 9 3900X 3.80 GHz CPU, 64 GB of RAM, and an Nvidia 4090 GPU with 24 GB of memory. The network is implemented with PyTorch. We use Adam optimizer to train the model for 200 epochs with a batch size of 16. The learning rate is set as 1e-5 and is divided by 10 at 160 and 180 epochs. Following previous methods [33, 45], data augmentation with random scaling ([-0.15, 0.15]) and random rotation ([-15, 15]) is applied to improve the model generalization ability. More details on implementation are provided in the supplementary materials.

4.3. Ablation Study

Following prior works [33, 45], we conduct experiments under split1 of MP-100. Due to computational resource constraints, we default to using 3-shot setting in ablation studies. We adopt the HRNet-32 pre-trained from ImageNet as the backbone. We first investigate the impact of cross-instance interaction on constructing support keypoint features. Then, we study the importance of dynamically activating for high-resolution features. Finally, we evaluate the impact of introducing support information in the iterative refinement process, including prior structure-based graph convolution and modulated deformable attention.

4.3.1 Cross-Instance Interaction

In this section, we compare three cross-instance interaction methods, including: 1) Average 2) Weighted Aggregation 3) Cross-instance Interaction. For weighted aggregation, we use the initial support keypoint features to predict the keypoint quality and use the predicted quality as the weight to aggregate support keypoint features. As shown in Table 1, direct averaging is susceptible to interference from low-quality keypoints and has the lowest performance. Adopting a weighted average can improve the quality of the aggregated keypoint features, thereby slightly improving the network's performance. Adopting the cross-instance interaction can better capture the complex dependencies between keypoints, thus achieving the best performance.

Table 1. Comparison of different aggregation methods. 'AW' and 'CI' stand for weighted aggregation and cross-instance interaction.

AW	CI	PCK
–	–	92.97
✓	–	93.21
✓	✓	**93.54**

Table 2. The Impact of dynamic activation. 'HR' and 'DC' stand for adopting high-resolution feature map and dynamic convolution.

HR	DC	PCK
–	–	92.84
✓	–	93.38
✓	✓	**93.54**

4.3.2 Dynamic Feature Activation

We demonstrate the effectiveness of leveraging dynamic convolution for high-resolution feature activation in Table 2. First, compared to using low-resolution features to update keypoint features, using high-resolution features to update has a better performance. This is because high-resolution features can maintain fine-grained visual information. Secondly, using dynamic convolution to reactivate high-resolution features can further improve the network performance, which shows that dynamic convolution operations can enhance category-related features and suppress irrelevant features in the query feature map.

4.3.3 Prior Structure and Graph Convolution

In this section, we explore the impact of different components in prior structure estimation and GCN. Specifically, we explore the impact of prior structure generation methods, sparse constraint, and graph convolution operation on network performance.

First, we compare three methods for estimating the prior structure, namely global regression, concatenation regression, and multi-head attention. The method based on global regression obtains a global keypoint feature through average pooling and then directly predicts an K^2-dimensional vector as a prior structure through an FC layer. Concatenation regression concat each keypoint feature with other keypoint features to construct an $K \times K \times 2C$-dimensional tensor and then performs element-wise connection strength prediction through an FC layer. As shown in Table 3, global regression (ID-1) is significantly worse than concatenation regression (ID-2) and multi-head attention (ID-6), while multi-head attention performs slightly better than concatenation regression. Therefore, we choose to use multi-head attention to predict the prior structure.

Then, we explore the impact of the sparsity constraint. As shown in Table 3, if the sparse constraint is not used

Table 3. Ablation study on different components in prior structure estimation and GCN. 'GR', 'CR' and 'MA' stand for global regression, concatenation regression, and multi-head attention. 'SC' stands for sparse constraint. 'SM', 'MM', and 'SEM' stand for shared matrix, modulation matrix, and self-information matrix.

ID	GR	CR	MA	SC	SM	MM	SEM	PCK
0								93.01
1	✓			✓			✓	93.17
2		✓		✓			✓	93.48
3			✓				✓	92.87
4			✓	✓	✓			93.15
5			✓	✓		✓		93.22
6			✓	✓			✓	**93.54**

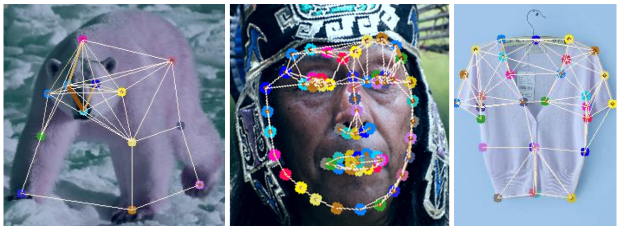


Figure 5. Visualization of estimated category-specific structures. We show the three categories of animals, faces and clothes. The yellow line represents the predicted edge between the keypoints.

(ID-3), a large number of meaningless edges are generated, severely reducing the effectiveness of the graph convolution operation. The performance of the network is even worse than not using GCN (ID-0). Finally, we explore the impact of different graph convolution operations. As shown in the Table 3, using a shared feature embedding matrix for all keypoints will limit the expressive capability of graph convolution (ID-4). Additionally, since keypoints of different categories have distinct attributes, adopting a fixed modulation matrix yields suboptimal results (ID-5). Generating the dynamic modulation matrix based on self-information of keypoints achieves the best performance.

At the same time, we visualize the estimated category-specific prior structure. As shown in Fig. 5, our method can generate reasonable prior structures for different categories. It can effectively capture the kinematic dependence between keypoints and some unique spatial relationships, such as the symmetrical relationship between the eyes. However, the structures generated for some objects with less structural characteristics still have certain defects. For example, the prior structure generated for clothing overlooks some connections on the overall silhouette, and the connections around the cuffs are disordered.

Table 4. Ablation study on different modulated deformable attention. 'SB' and 'DB' stand for support-based generation and difference-based generation. 'UM' and 'SM' stand for unique modulation coefficient and shared modulation coefficient.

SB	DB	UM	SM	PCK
				93.17
✓			✓	93.06
	✓	✓		93.32
	✓		✓	**93.54**

4.3.4 Modulated Deformable Attention

In this section, we compared two methods for generating modulation coefficients and two ways of modulating offsets. First, inspired by dynamic convolution, we use the semantic-aware support keypoint feature to generate modulation coefficients. As shown in Table 4, this method performs worse than standard deformable attention. We argue that this is because the semantic-aware support keypoint feature lacks query information and cannot effectively assess the quality of the query keypoints, thus failing to modulate the offset. Secondly, we tried generating a unique modulation coefficient for each offset of the same keypoint. As shown in Table 4, compared to generating a shared modulation coefficient for all offsets of the same keypoint, adopting a shared modulation coefficient brought only a minor performance improvement. This indicates that giving excessive freedom to sparse feature sampling could be harmful.

4.4. Comparisons with SOTA

We compare SIANet with the previous CAPE method, including POMNet [45], CapeFormer [33], ProtoNet [34], MAML [13] and Fine-tune [25]. For fairness of comparison, we also report the performance of SPDNet using ResNet-50 as the backbone, and accordingly abandon the dynamic modulation designed for high-resolution features. As shown in Table 5, SDPNet outperforms the SOTA method CapeFormer [33] under both 1-shot and 5-shot settings. Specifically, our method improves the average PCK by 2.05% and 1.12% in 1-shot setting and 5-shot setting.

We show some qualitative results comparisons with SOTA methods. As shown in the Fig. 6, our method demonstrates some unique advantages. Firstly, SDPNet can focus on fine-grained visual information, achieving more detailed and precise predictions. For example, in the prediction of the hind legs of the dog and fox in the first row, SPDNet can achieve pixel-aligned keypoint estimation. This is attributed to high-resolution feature sampling based on modulated deformable attention. Secondly, SPDNet is highly robust and can handle variations in local texture, color, and shape. For example, in the second row on the left, due to the interference of hair, the texture and shape of the eye area have

Table 5. Comparisons with the SOTA methods on MP-100 dataset under both 1-shot and 5-shot settings (PCK).

Method	1-Shot						5-shot					
	Split 1	Split 2	Split 3	Split 4	Split 5	Average	Split 1	Split 2	Split 3	Split 4	Split 5	Average
ProtoNet [34]	46.05	40.84	49.13	43.34	44.54	44.78	60.31	53.51	61.92	58.44	58.61	58.56
MAML [13]	68.14	54.72	64.19	63.24	57.20	61.50	70.03	55.98	63.21	64.79	58.47	62.50
Fine-tune [25]	70.60	57.04	66.06	65.00	59.20	63.58	71.67	57.84	66.76	66.53	60.24	64.61
POMNet [45]	84.23	78.25	78.17	78.68	79.17	79.70	84.72	79.61	78.00	80.38	80.85	80.71
CapeFormer [33]	89.45	84.88	83.59	83.53	85.09	85.31	91.94	88.92	89.40	88.01	88.25	89.30
SDPNet(ResNet-50)	**90.03**	**85.42**	**84.22**	**84.17**	**85.95**	**85.96**	**92.65**	**89.64**	**89.46**	**88.57**	**88.74**	**89.81**
SDPNet(HRNet-32)	**91.54**	**86.72**	**85.49**	**85.77**	**87.26**	**87.36**	**93.68**	**90.23**	**89.67**	**89.08**	**89.46**	**90.42**

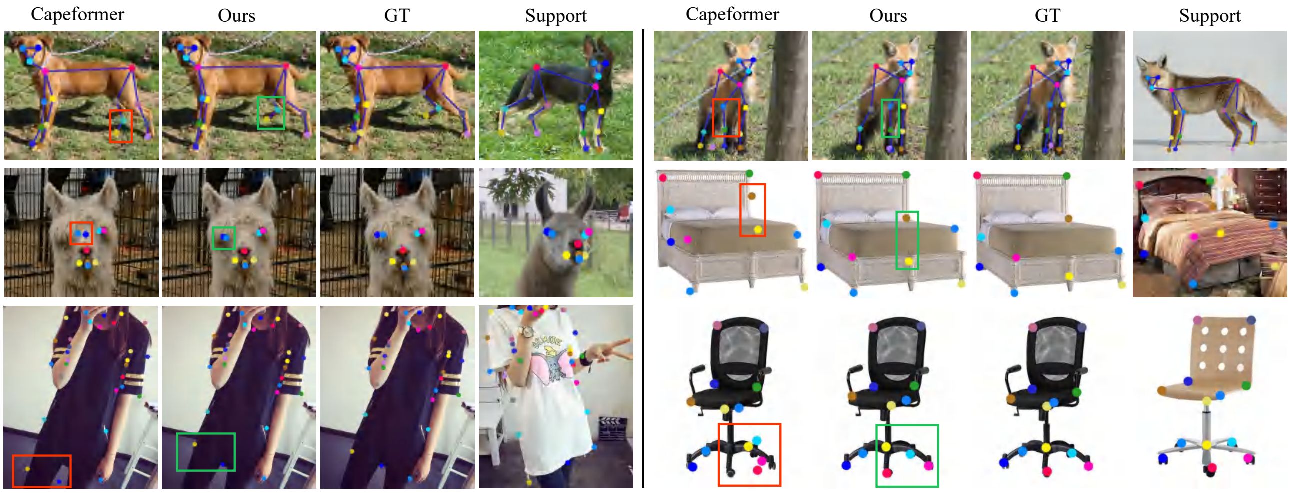


Figure 6. Qualitative results. We visualize the keypoint predictions under 5-shot setting. For simplicity, we only show one support image. The bones are not the results predicted by our network, but provided by the dataset.

significant changes. However, our method can still accurately predict the position of the eye keypoints; in the third row on the left, there are considerable changes in the color and angle of the clothing, yet our method can predict the position of the keypoints accurately. Finally, our method is also robust to occlusions. In the second and third rows on the right, some keypoints of the bed and chair are not visible due to self-occlusion. CapeFormer [33] cannot predict these occluded keypoints, but SDPNet can locate these keypoints. We attribute the robustness of SPDNet to the dynamic graph convolution, which can perform information passing based on the estimated prior structures.

5. Conclusion

In this paper, we propose a Support-based Dynamic Perception Network (SDPNet) for the robust and accurate category-agnostic pose estimation. SDPNet fully leverages the visual feature and keypoint information of support samples to guide the query pose estimation. SDPNet utilizes the semantic information of support samples to dynamically activate the query visual feature map and modulate the sampling process of the query keypoint feature. At the same time, SDPNet uses the support keypoint information to dynamically predict category-specific prior structures, and adopts graph convolution to perform structured information interaction between query keypoints. SDPNet can achieve pixel-aligned pose estimation and is robust to local appearance changes. However, SPDNet does not explicitly consider the 3D structure of the object, resulting in unsatisfactory prediction accuracy for some occluded keypoints.

Acknowledgement. This work was supported by the National Natural Science Foundation of China under Grants (62201072, 62171057, 62101064, U23B2001, 62001054, 62071067), the National Postdoctoral Program for Innovative Talents under Grant BX20230052, the China Postdoctoral Science Foundation (2023TQ0039), the Postdoctoral Fellowship Program of CPSF (GZC20230320), the Ministry of Education and China Mobile Joint Fund (MCM20200202, MCM20180101), Beijing University of Posts and Telecommunications-China Mobile Research Institute Joint Innovation Center.

References

[1] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. *NIPS*, 35:25005–25017, 2022.

[2] Luca Bertinetto, João F Henriques, Jack Valmadre, Philip H. S Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *NeurIPS*, 2016.

[3] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? In *ICCV*, pages 1021–1030, 2017.

[4] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *ICCV*, pages 2272–2281, 2019.

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, pages 7291–7299, 2017.

[6] Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David J Fleet, and Geoffrey E Hinton. A unified sequence interface for vision tasks. *NIPS*, 35:31333–31346, 2022.

[7] Zheng Chen, Sihan Wang, Yi Sun, and Xiaohong Ma. Self-supervised transfer learning for hand mesh recovery from binocular images. In *ICCV*, pages 11626–11634, 2021.

[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 *CVPR*, 2020.

[9] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *CVPR*, pages 183–192, 2020.

[10] Shiyang Cheng, Georgios Tzimiropoulos, Jie Shen, and Maja Pantic. Faster, better and more detailed: 3d face reconstruction with graph convolutional networks. In *Proceedings of the Asian conference on computer vision*, 2020.

[11] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J Crandall. Hope-net: A graph-based model for hand-object pose estimation. In *CVPR*, pages 6608–6617, 2020.

[12] Linpu Fang, Xingyan Liu, Li Liu, Hang Xu, and Wenxiong Kang. Jgr-p2o: Joint graph reasoning based pixel-to-offset prediction network for 3d hand pose estimation from a single depth image. In *ECCV*, pages 120–137. Springer, 2020.

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. 2017.

[14] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *CVPR*, pages 10833–10842, 2019.

[15] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Han Hu, Dong Chen, et al. Instructdiffusion: A generalist modeling interface for vision tasks. *arXiv preprint arXiv:2309.03895*, 2023.

[16] Tanmay Gupta, Amita Kamath, Aniruddha Kembhavi, and Derek Hoiem. Towards general purpose vision systems: An end-to-end task-agnostic vision-language architecture. In *CVPR*, pages 16399–16409, 2022.

[17] Weiting Huang, Pengfei Ren, Jingyu Wang, Qi Qi, and Haifeng Sun. Awr: Adaptive weighting regression for 3d hand pose estimation. In *AAAI*, pages 11061–11068, 2020.

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456. pmlr, 2015.

[19] Muhammad Haris Khan, John McDonagh, Salman Khan, Muhammad Shahabuddin, Aditya Arora, Fahad Shahbaz Khan, Ling Shao, and Georgios Tzimiropoulos. Animalweb: A large-scale hierarchical dataset of annotated animal faces. In *CVPR*, 2020.

[20] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, pages 4990–5000, 2020.

[21] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. In *ICCV*, 2021.

[22] 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 *ECCV*, 2014.

[23] Kenkun Liu, Rongqi Ding, Zhiming Zou, Le Wang, and Wei Tang. A comprehensive study of weight sharing in graph networks for 3d human pose estimation. In *ECCV*, pages 318–334. Springer, 2020.

[24] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv:2206.08916*, 2022.

[25] Akihiro Nakamura and Tatsuya Harada. Revisiting fine-tuning for few-shot learning. *arXiv preprint arXiv:1910.00216*, 2019.

[26] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[27] Xuecheng Nie, Jiashi Feng, Yiming Zuo, and Shuicheng Yan. Human pose estimation with parsing induced learner. In *CVPR*, pages 2100–2108, 2018.

[28] Markus Oberweger and Vincent Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *ICCV Workshops*, pages 585–594, 2017.

[29] Pengfei Ren, Haifeng Sun, Jiachang Hao, Jingyu Wang, Qi Qi, and Jianxin Liao. Mining multi-view information: a strong self-supervised framework for depth-based 3d hand pose and mesh estimation. In *CVPR*, pages 20555–20565, 2022.

[30] Pengfei Ren, Chao Wen, Xiaozheng Zheng, Zhou Xue, Haifeng Sun, Qi Qi, Jingyu Wang, and Jianxin Liao. Decoupled iterative refinement framework for interacting hands reconstruction from a single rgb image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8014–8025, 2023.

[31] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, pages 12026–12035, 2019.

[32] Lei Shi, Yifan Zhang, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *CVPR*, pages 7912–7921, 2019.

[33] Min Shi, Zihao Huang, Xianzheng Ma, Xiaowei Hu, and Zhiguo Cao. Matching is not enough: A two-stage framework for category-agnostic pose estimation. In *CVPR*, 2023.

[34] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *NeurIPS*, 2017.

[35] Xibin Song, Peng Wang, Dingfu Zhou, Rui Zhu, Chenye Guan, Yuchao Dai, Hao Su, Hongdong Li, and Ruigang Yang. Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In *CVPR*, 2019.

[36] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33(5):169:1–169:10, 2014.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[38] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dense 3d regression for hand pose estimation. In *CVPR*, pages 5147–5156, 2018.

[39] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *ICML*, pages 23318–23340. PMLR, 2022.

[40] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *CVPR*, pages 6830–6839, 2023.

[41] Yangang Wang, Cong Peng, and Yebin Liu. Mask-pose cascaded cnn for 2d hand pose estimation from single color image. *TCSVT*, 2018.

[42] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[43] Yue Wu and Qiang Ji. Facial landmark detection: A literature survey. *IJCV*, 127:115–142, 2019.

[44] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the ECCV*, pages 466–481, 2018.

[45] Lumin Xu, Sheng Jin, Wang Zeng, Wentao Liu, Chen Qian, Wanli Ouyang, Ping Luo, and Xiaogang Wang. Pose for everything: Towards category-agnostic pose estimation. In *ECCV*. Springer, 2022.

[46] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

[47] Hang Yu, Yufei Xu, Jing Zhang, Wei Zhao, Ziyu Guan, and Dacheng Tao. Ap-10k: A benchmark for animal pose estimation in the wild. *arXiv preprint arXiv:2108.12617*, 2021.

[48] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *CVPR*, pages 7093–7102, 2020.

[49] Yuxiang Zhang, Liang An, Tao Yu, Xiu Li, Kun Li, and Yebin Liu. 4d association graph for realtime multi-person motion capture using multiple video cameras. In *CVPR*, pages 1324–1333, 2020.

[50] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *CVPR*, pages 3425–3435, 2019.

[51] Weixi Zhao, Weiqiang Wang, and Yunjie Tian. Graformer: Graph-oriented transformer for 3d pose estimation. In *CVPR*, pages 20438–20447, 2022.

[52] Yanping Zheng, Guang Zeng, Haisheng Li, Qiang Cai, and Junping Du. Colorful 3d reconstruction at high resolution using multi-view representation. *JVCIR*, 2022.

[53] 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.

[54] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *ICCV*, 2017.

[55] Zhiming Zou and Wei Tang. Modulated graph convolutional network for 3d human pose estimation. In *ICCV*, pages 11477–11487, 2021.

[56] Zhiming Zou, Kenkun Liu, Le Wang, and Wei Tang. High-order graph convolutional networks for 3d human pose estimation. In *BMVC*, 2020.