

Recurrent Feature Mining and Keypoint Mixup Padding for Category-Agnostic Pose Estimation

Junjie Chen¹ Weilong Chen¹ Yifan Zuo¹ Yuming Fang^{1*}

¹ Jiangxi University of Finance and Economics

{chenjunjie, chenweilong, zuoyifan, fangyuming}@jxufe.edu.cn

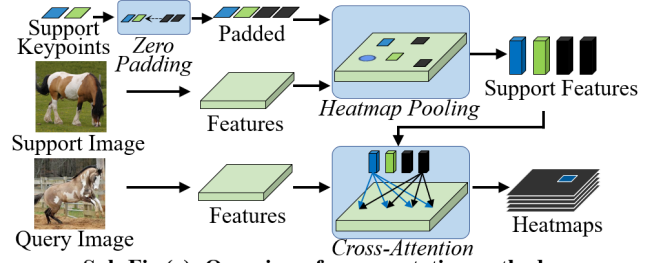
Abstract

Category-agnostic pose estimation aims to locate keypoints on query images according to a few annotated support images for arbitrary novel classes. Existing methods generally extract support features via heatmap pooling, and obtain interacted features from support and query via cross-attention. Hence, these works neglect to mine fine-grained and structure-aware (FGSA) features from both support and query images, which are crucial for pixel-level keypoint localization. To this end, we propose a novel yet concise framework, which recurrently mines FGSA features from both support and query images. Specifically, we design a FGSA mining module based on deformable attention mechanism. On the one hand, we mine fine-grained features by applying deformable attention head over multi-scale feature maps. On the other hand, we mine structure-aware features by offsetting the reference points of keypoints to their linked keypoints. By means of above module, we recurrently mine FGSA features from support and query images, and thus obtain better support features and query estimations. In addition, we propose to use mixup keypoints to pad various classes to a unified keypoint number, which could provide richer supervision than the zero padding used in existing works. We conduct extensive experiments and in-depth studies on large-scale MP-100 dataset, and outperform SOTA method dramatically (+3.2%PCK@0.05).

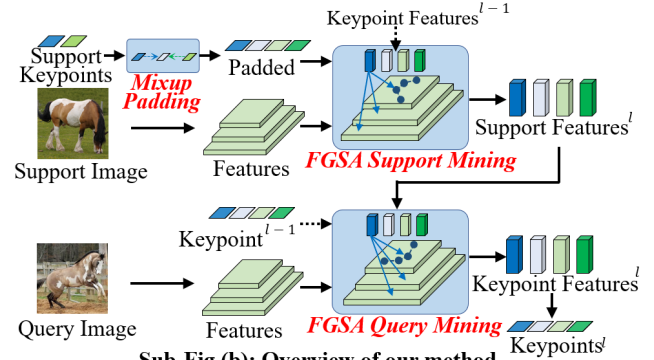
1. Introduction

Pose estimation is a fundamental and significant computer vision task, which aims to produce the locations of pre-defined semantic part of object instance in 2D image. Recently, it has received increasing attention in the computer vision community due to its wide applications in virtual reality, augmented reality, human-computer interaction, robot and automation. However, most pose estimation methods are trained with category-specific data and thus cannot be

*Corresponding author



Sub-Fig (a): Overview of representative methods.



Sub-Fig (b): Overview of our method.

Figure 1. Overview of representative methods and ours. (a): Representative methods use zero padding to align keypoint number, and rely on the heatmap pooling and cross-attention on single-scale features to produce support features and query estimations. (b): Our method uses mixup padding, and recurrently mines FGSA features from multi-scale features of support and query images, which can produce better support features and query estimations.

applied for novel classes, especially when they have different keypoint classes. Therefore, category-agnostic pose estimation (CAPE) [52] is introduced to locate target keypoints for arbitrary classes given a few support images annotated with keypoints.

In CAPE, the query images come from novel classes, and thus the target keypoints to be estimated are determined by the annotated keypoints on support images. Consequently, extracting high-quality features from both support images and query images are fundamental issues in CAPE. As

shown in Fig. 1 (a), representative methods [45, 52] firstly extract support features via feature map pooling weighted by a heatmap drawing annotated keypoint, and then enable the interaction between support features and query feature map by cross-attention. However, the pooling or cross-attention over single feature map are coarse-grained and inadequate for pixel-level keypoint localization. Recent methods propose to complement support features with prototypical [8, 37], global [30] or structural [19, 42] information, but how to extract fine-grained and structure-aware features from both support and query images remains unexplored.

In this paper, we propose a novel yet concise framework to recurrently mines fine-grained and structure-aware (FGSA) features from both support and query images. Specifically, our framework consists of multiple stacked layers, and the pipeline of each layer is shown in Fig. 1 (b). Generally, each layer firstly extracts FGSA features from the feature pyramid of support images according to the given support keypoints, and then absorbs FGSA features from query images to produce high-quality keypoint features. In this way, our framework could recurrently refine the support features, keypoint features and estimated keypoints, and thus estimate more precise keypoints.

To mine FGSA features from both support and query images, we design a FGSA mining module based on deformable attention [63], which is more flexible and effective against the heatmap pooling or cross-attention used in [30, 45, 52]. On the one hand, we set keypoints as reference points to mine fine-grained features from feature pyramid, which is especially beneficial for pixel-level keypoint localization. On the other hand, we offset the attention heads of each keypoint to its linked keypoints, which could adaptively extract structure-aware features to facilitate the keypoint localization. Therefore, our module could produce high-quality support features and keypoint features.

Besides, an inevitable issue in CAPE is keypoint padding, because all classes could have different numbers of keypoints. Previous works intuitively align the keypoint number of various classes by zero padding, *i.e.*, generating meaningless keypoints with zero weight/visibility. Inspired by Mixup [58] and DensePose [17], we propose to align the keypoint number by mixup padding, *i.e.*, generating dense keypoints by random mixing two linked keypoints. Therefore, our mixup padding could provide richer supervision and enable our model to learn denser semantics of poses.

For the experimental setting, we follow previous works [8, 45, 52] to conduct experiments on MP-100 dataset [52]. The quantitative and qualitative experiments demonstrate the effectiveness of our proposed method. Our contributions can be summarized as: **1)** We propose a novel framework to recurrently mine fine-grained and structure-aware features from both support and query images, which could extract better support features and locate more precise key-

points on query images. **2)** We propose a feature mining module based on deformable attention to integrally mine fine-grained and structure-aware features. **3)** We propose a mixup padding strategy to provide richer supervision for model to learn denser semantics. **4)** We conduct comprehensive experiments on MP-100 dataset, and our model outperforms SOTA method dramatically (+3.2% PCK@0.05).

2. Related Works

2.1. Category-Specific Pose Estimation

Pose estimation is a fundamental and important vision task, aiming at detecting pre-defined keypoints of objects in image. Most existing methods are class-specific, *i.e.*, focusing on estimating keypoints for single and specific class, *e.g.*, humans [1, 27], animals [4, 26], or vehicles [41, 46]. Technically, existing pose estimation approaches can be broadly divided into three groups: heatmap-based methods [9, 10, 57], regression-based methods [16, 28, 38], and query-based methods [36, 44, 53]. For example, PETR[44] introduced multiple pose queries to reason about a set of full-body poses, with a joint decoder to refine using kinematic relationships. RLE [28] is a novel regression paradigm using Residual Log-likelihood Estimation to capture distributional changes and could facilitate the training process. SWAHR [35] adopted scale-adaptive heatmap estimation, which adjusts the standard deviation for each keypoint adaptively, making it more robust to varying scales and ambiguities. DEKR [16] designed a multi-branch structure for disentangled keypoint regression, which enables the model to focus on keypoint regions and improve performance. Although existing methods have achieved great success in locating keypoints for specific classes, they cannot directly be applied for novel classes, particularly when the target classes have different numbers or types of keypoints.

2.2. Category-Agnostic Pose Estimation

Transfer learning is effective to learn novel classes, and thus various methods [2, 6, 7, 21, 31, 34, 43, 54, 55] have been proposed for extending the class scope for pose estimation models. Specifically in few-shot learning, previous methods have primarily concentrated on specific domains, such as facial images [3, 50], clothing images [15], or animal images [47, 59–61]. For more diverse categories, POMNet [52] elaborated a large-scale dataset including 100 classes and introduced a keypoint matching framework to locate target keypoints. CapeFormer [45] improved similarity modeling within above matching pipeline and further refined of each keypoint using a sophisticated transformer decoder. SCAPE [30] proposed global keypoint feature perceptor and keypoint attention refiner to locate target keypoints with self-attention layers. Besides, Lu *et al.* [31] introduced a flexible few-shot scenario that includes both

novel/base classes and novel/base keypoints. Although existing methods [32, 33, 37, 40, 42] have greatly advanced CAPE, how to extract FGSA features from both support and query images and how to better padding keypoints remain unexplored. In this paper, we propose a recurrent framework to flexibly use deformable attention to mine FGSA features from support and query images, and also propose keypoint mixup padding for CAPE.

2.3. Attention Mechanism

Attention mechanism has been widely applied in vision tasks and achieved great success. Existing attention mechanism could be roughly categorized into two prevalent forms: importance weight and spatial transformation. Importance weight is a straightforward form of attention, including spatial attention[24, 49], channel attention [20], and the combination of both[14, 22, 29, 51]. Spatial transformation is a special form of attention as discussed in [23, 62], *e.g.*, Spatial Transformer Network [23] and Deformable ConvNet (DCN) [12]. Recently, multi-head attention [48] has shown remarkable effects in a wide range of vision tasks, but suffers from efficiency issues when applied to high-resolution feature maps. To this end, deformable attention [5] uses spatial transformation to effectively mine features around reference points, enabling modules to mine fine-grained features on multi-scale feature maps. In this paper, we design our feature miner upon deformable attention module [5] to recurrently mine fine-grained features from support and query images, and meanwhile set reference points according to links to extract structure-aware features.

3. Method

In this section, we first formally describe the task setting of class-agnostic pose estimation (CAPE) in Sec. 3.1. Afterwards, we introduce the overall pipeline of our framework in Sec. 3.2. Then, we respectively introduce the details of FGSA mining module in Sec. 3.3 and keypoint mixup padding and Sec. 3.4. Finally, we describe the training and inference pipeline in Sec. 3.5.

For brevity of description, we use non-bold letter to denote scalar, and employ bold letter to represent vector/matrix/tensor. We adopt subscript to indicate the variable source and use square bracket to show the index in variable, *e.g.*, $\mathbf{P}_s[k]$ means the k -th keypoint on support image. Additionally, we employ $[\cdot; \cdot]$ to represent the concatenation of two variables.

3.1. Task Setting

Category-agnostic pose estimation (CAPE) aims to locate target keypoints of query image for any novel category, where the target keypoints are determined by a few annotated support images of the same category. Formally, N -shot setting means there are N support images available,

which could be formulated as:

$$\mathbf{P}_q = f_{cape}(\mathbf{x}_q, \{\mathbf{x}_{s_n}\}_{n=1}^N, \{\mathbf{P}_{s_n}^*\}_{n=1}^N, \mathbf{L}_c^*), \quad (1)$$

where $\mathbf{x}_q \in \mathbb{R}^{H \times W \times 3}$ and $\{\mathbf{x}_{s_n}\}_{n=1}^N$ are the query image and N support images from category c . The target keypoints to be estimated $\mathbf{P}_q$ and the n -th support keypoints $\mathbf{P}_{s_n}$ have the same size, *i.e.*, $\mathbf{P}_q \in \mathbb{R}^{K_c \times 2}$, where K_c is the keypoint number of category c . Additionally, $\mathbf{L}_c^* \in \mathbb{R}^{K_c \times K_c}$ is a binary matrix to link keypoints to pose of category c , *i.e.*, $\mathbf{L}_c^*[i, j]$ means the link existence between the i -th and the j -th keypoints.

To learn and evaluate above function $\mathcal{F}_{cape}$, all categories in benchmark are split into base categories $\mathcal{C}^b$ and novel categories $\mathcal{C}^n$, where $\mathcal{C}^b \cap \mathcal{C}^n = \emptyset$. In the training stage, the query and support images come from only base classes, *i.e.*, $c \in \mathcal{C}^b$. In the test stage, images come from novel classes, *i.e.*, $c \in \mathcal{C}^n$. For simplicity, we first describe our framework in 1-shot setting, *i.e.*, $\mathbf{P}_q = f_{cape}(\mathbf{x}_q, \mathbf{x}_s, \mathbf{P}_s^*, \mathbf{L}_c^*)$, and then introduce the extension to multiple support images.

3.2. Overall Pipeline

Considering that CAPE is a pixel-level estimation task and the target keypoints are determined by both query and support images, our proposed framework recurrently mines fine-grained and structure-aware (FGSA) features from both support and query images, and thus could estimate more precise target keypoints on query images.

Specifically, our framework consists of L stacked layers, and the detailed architecture of single layer is illustrated in Fig. 2. For the l -th layer, we denote the support features as $\mathbf{F}_s^l \in \mathbb{R}^{K \times D}$, which contain the semantic information of K target keypoints extracted from support image and support keypoints. We denote the keypoint features as $\mathbf{F}_q^l \in \mathbb{R}^{K \times D}$, which store the specific information of K target keypoints on query image determined by support features. The recurrent pipeline of updating support features $\mathbf{F}_s^l$ and keypoint features $\mathbf{F}_q^l$ are described as follows.

Given the support image $\mathbf{x}_s$, support keypoints $\mathbf{P}_s^*$, category-wise link $\mathbf{L}_c^*$ and query image $\mathbf{x}_q$, our framework firstly use mixup padding strategy to align the keypoint number K_c to a unified number K , formulated as:

$$[\mathbf{P}_s; \mathbf{L}_c] = f_{mixup}(\mathbf{P}_s^*, \mathbf{L}_c^*, \alpha), \quad (2)$$

where α is the mixup parameter as in [58], and $\mathbf{P}_s \in \mathbb{R}^{K \times 2}$ and $\mathbf{L}_c \in \mathbb{R}^{K \times K}$ are the padded keypoints and links. The details about our proposed f_{mixup} reminds to Sec. 3.4. Meanwhile, our framework uses a backbone network to extract multi-scale feature maps (pyramid) from support and query images, denoted as $\mathcal{F}_s$ and $\mathcal{F}_q$ respectively.

Afterwards, our framework produces support features $\mathbf{F}_s^l \in \mathbb{R}^{K \times D}$ by complementing previous keypoint features

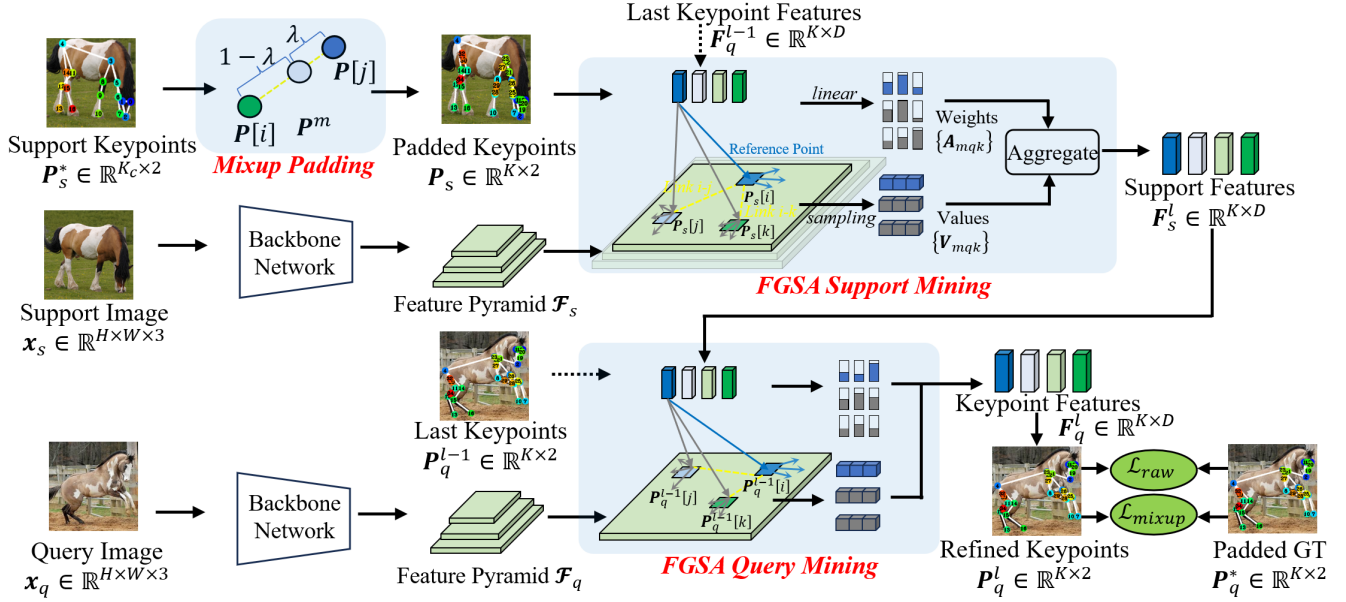


Figure 2. The architecture of our l -th recurrent layer. Firstly, support keypoints P_s^* is aligned by mixup padding, which produces padded keypoints P_s . Then, $f_{miner-s}$ mines FGSA features around P_s on support image to extract support features F_s^l according to the last keypoint features F_q^{l-1} . Afterwards, $f_{miner-q}$ mines FGSA features around the last keypoints P_q^{l-1} on query image to extract keypoints F_q^l , which produces target keypoints P_q^l . By recurrently updating F_s^l , F_q^l and P_q^l , our framework finally estimates precise keypoints P_q^L .

F_q^{l-1} with the FGSA features mined from support feature pyramid $\mathcal{F}_s$, which could be summarized as:

$$F_s^l = f_{miner-s}(F_q^{l-1}, \mathcal{F}_s, P_s, L_c), \quad (3)$$

where $f_{miner-s}$ denotes our proposed module mining FGSA features based on support keypoints P_s and category-wise links L_c , and the details remain to Sec. 3.3. In this way, the support features F_s^l are also extracted according to keypoints features F_q^{l-1} , and thus provide more targeted support information to refine target keypoints.

Based on the extracted support features F_s^l , our framework uses another FGSA feature miner $f_{miner-q}$ to produce keypoint features $F_q^l \in \mathbb{R}^{K \times D}$ by retrieving matched FGSA features with in query feature pyramid $\mathcal{F}_q$, as:

$$F_q^l = f_{miner-q}(F_s^l, \mathcal{F}_q, P_q^{l-1}, L_c), \quad (4)$$

where $f_{miner-q}$ mines FGSA features based on the last keypoints P_q^{l-1} and category-wise links L_c . In the l -th layer, the target keypoints $P_q^l \in \mathbb{R}^{K \times 2}$ are estimated by:

$$P_q^l = \sigma(\sigma^{-1}(P_q^{l-1}) + f_{mlp}(F_q^l)), \quad (5)$$

where σ is the Sigmoid function, f_{mlp} is a light-weight MLP, and the keypoint predicting follows the incremental refinement in previous works [8, 45].

To launch our recurrent framework, we set F_q^0 using the heatmap pooled features of P_s and $\mathcal{F}_s$, and set P_q^0 using the

mid-value 0.5. By recurrently applying Eqn. 3, Eqn. 4 and Eqn. 5, our framework recurrently updates F_s^l , F_q^l and P_q^l , and finally obtains precise target keypoints, i.e., P_q^L . The module details are introduced in following sub-sections.

3.3. FGSA Feature Mining

To mine fine-grained and structure-aware (FGSA) features to benefit CAPE task, we design a universal module based on deformable attention [5], which could be applied on both support and query images.

As a flexible module, each attention head in deformable attention [5] learns multiple sampling offsets to adaptively aggregate features from multi-scale feature maps $\mathcal{F}$ around the reference point $p \in \mathbb{R}^2$. We formulate the function of single attention head as:

$$f' = f_{att}(f, \mathcal{F}, p), \quad (6)$$

where f and $f' \in \mathbb{R}^D$ are the query and output feature vector. The internal details (e.g., sampling offsets and attention weights) could be found in [5], and thus omitted for brevity. Based on Eqn. 6, our module naturally mines fine-grained features from feature pyramid using support keypoints or target keypoints as reference points.

The original attention module [5] uses identical reference point for multiple attention heads, while we propose to use L -adapted reference points to capture structure-aware features. As aforementioned, our module is summarized as

$f_{miner}(\mathbf{F}, \mathcal{F}, \mathbf{P}, \mathbf{L})$, where $\mathbf{F} \in \mathbb{R}^{K \times D}$, $\mathbf{P} \in \mathbb{R}^{K \times 2}$ and $\mathbf{L} \in \mathbb{R}^{K \times K}$ denote queries, keypoints and links. For the M attention heads of k -th query, the respective M reference points $\mathcal{P}_k \in \mathbb{R}^{M \times 2}$ are derived via Breadth-First Search in the graph defined by $\mathbf{L}$ and starting point $\mathbf{P}[k]$. And the process of our f_{miner} for the k -th query is:

$$\mathbf{F}'[k] = \mathbf{F}[k] + \sum_m^M \mathbf{W}_m \cdot f_{att[m]}(\mathbf{F}[k], \mathcal{F}, \mathcal{P}_k[m]), \quad (7)$$

which can naturally extend to K queries $\mathbf{F} \in \mathbb{R}^{K \times 2}$ as [5].

Therefore, our module flexibly utilizes deformable attention heads to extract fine-grained and structure-aware features, and thus can benefit the feature extraction in both support and query images, *i.e.*, Eqn. 3 and Eqn. 4.

3.4. Keypoint Mixup Padding

In CAPE, different classes could have different numbers of keypoints, and thus keypoint padding for aligning keypoint number is inevitable. Inspired by Mixup [58] and DensePose [17], we propose keypoint mixup padding, which could provide richer supervision and denser semantics than the zero padding used in existing works [8, 45, 52].

As aforementioned in Eqn. 2, our f_{mixup} pads K_c keypoints to K keypoints according to class-wise link $\mathbf{L}_c^*$. Firstly, we random sample $K - K_c$ keypoint pairs from all linked keypoint pairs. Afterwards, for each sampled keypoint pair $\mathbf{P}^*[i]$ and $\mathbf{P}^*[j]$, we individually sample a $\lambda \sim \text{Beta}(\alpha, \alpha)$ to mix them up:

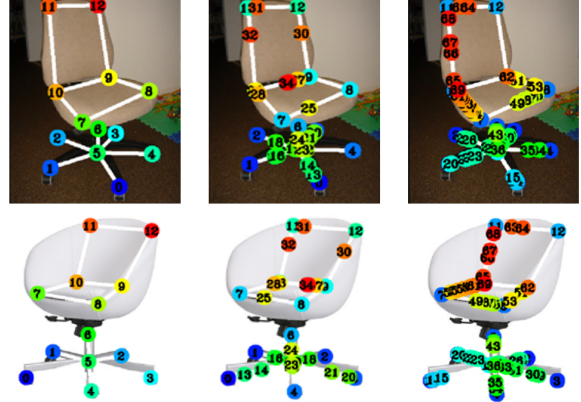
$$\mathbf{P}[k] = \lambda \cdot \mathbf{P}^*[i] + (1 - \lambda) \cdot \mathbf{P}^*[j], \quad (8)$$

where $\mathbf{P}[k]$ denotes the padded keypoint. Then, we collect all padded keypoints on the same link, and sequentially link them to replace the original link, *e.g.*, $N + 1$ new links for N padded keypoints. Finally, we obtain the padded keypoints $\mathbf{P} \in \mathbb{R}^{K \times 2}$ and links $\mathbf{L} \in \mathbb{R}^{K \times K}$.

In the training stage, the support keypoints and GT target keypoints should keep consistent, and thus we apply the same λ in each keypoint mixup. In the inference stage, we use uniform padding by generating equal division points on various links for a stable output. The examples of keypoint mixup padding are illustrated in Fig. 3 using $K = 35$ or $K = 70$ with $\alpha = 1.0$. As we can see, the padded keypoints generally distribute appropriately over the object structures, and the padded support keypoints and padded target keypoints on query image also keep the semantic consistency between support and query images. Therefore, our model could learn denser semantics to facilitate CAPE.

3.5. Training and Inference

By our framework (Sec. 3.2) and internal modules (Sec. 3.3 and Sec. 3.4), we obtain L target keypoints $\{\mathbf{P}_q^l\}_{l=1}^L$ for



(1) Raw keypoints. (2) Mixup to 35 keypoints. (3) Mixup to 70 keypoints.

Figure 3. Illustration of keypoint mixup padding, *i.e.*, padding 16 support keypoints (upper) and target keypoints (bottom) to 35 or 70 keypoints. Therefore, our padding strategy could provide richer supervision and enable model to learn denser semantics.

query image x_q . Our full training objective is twofold:

$$\mathcal{L}_{full} = \mathcal{L}_{raw} + \beta \cdot \mathcal{L}_{mixup}, \quad (9)$$

where β is a hyper-parameter for balancing. Specifically, the first objective $\mathcal{L}_{raw}$ supervises the predictions corresponding to raw keypoints as in previous work [45]:

$$\mathcal{L}_{raw} = \frac{1}{L} \sum_{l=1}^L \sum_{k=1}^{K_c} \left| \mathbf{P}_q^l[k] - \mathbf{P}_q^*[k] \right|_1, \quad (10)$$

where K_c is the raw keypoint number before padding, and $\mathbf{P}_q^*$ denotes the GT keypoints after padding. Similarly, the second objective $\mathcal{L}_{mixup}$ is:

$$\mathcal{L}_{mixup} = \frac{1}{L} \sum_{l=1}^L \sum_{k=K_c+1}^K \left| \mathbf{P}_q^l[k] - \mathbf{P}_q^*[k] \right|_1, \quad (11)$$

which supervises on the $K - K_c$ padded keypoints.

In inference, our model outputs the result of the last layer $\mathbf{P}_q^L$ as the final estimation. In N -shot setting, we mine FGSA features on N support images in each recurrent layer and average N mined features to obtain support features, *i.e.*, $\mathbf{F}_s^l = \frac{1}{N} \sum_{n=1}^N \mathbf{F}_{s,n}^l$.

4. Experiments

4.1. Dataset, Metric, and Implementation Details

Following previous CAPE studies, we utilize the MP-100 dataset [52] for both training and evaluation. This dataset spans 100 classes organized into 8 super-classes, making it the largest benchmark dataset for category-aware pose estimation (CAPE). MP-100 includes samples drawn from various category-specific pose estimation datasets, containing

Table 1. The mPCK (% $\uparrow$) performances of various methods in 1-shot setting and 5-shot setting on 5 splits of MP-100 dataset. We summarize the PCK results with different threshold in Tab. 2. The best results are highlighted using boldface.

Method	1-shot setting						5-shot setting					
	Split1	Split2	Split3	Split4	Split5	AVG	Split1	Split2	Split3	Split4	Split5	AVG
POMNet [52]	69.22	63.23	62.87	63.42	63.92	64.53	71.31	67.59	66.41	67.78	68.31	68.28
CapeFormer [45]	75.13	69.30	68.59	68.50	71.38	70.58	78.05	74.43	74.87	73.80	76.11	75.45
ESCAPE [37]	72.42	66.83	63.92	64.71	64.48	66.47	75.53	73.18	71.12	72.24	72.68	72.95
MetaPoint [8]	77.11	71.07	70.32	69.93	72.73	72.23	79.22	75.51	76.20	75.92	77.65	76.90
GraphCape [19]	73.47	68.45	67.61	67.32	68.02	68.97	77.21	74.11	73.25	73.28	73.09	74.19
SCAPE [30]	77.53	71.21	70.41	69.51	73.12	72.36	79.35	76.26	76.29	76.12	77.89	77.18
Ours	78.72	72.32	71.38	70.74	73.93	73.42	80.67	77.24	76.92	77.13	78.12	78.02

over 18K images and 20K annotations, with the number of keypoints varying from 8 to 68 across different classes. The 100 classes in MP-100 are divided into non-overlapping training, validation, and test sets in a 70 : 10 : 20 ratio. To ensure that evaluation categories remain unseen during training, the dataset is further split into five mutually exclusive partitions, each maintaining this separation throughout the training and testing process. By default, we use $K = 70$, $M = 8$, $L = 3$, $\alpha = 1$ and $\beta = 0.5$ for our model. We use Probability of Correct Keypoint (PCK) [56] as the quantitative metric. Apart from the PCK of threshold 0.2 as in [45, 52], we also report the mPCK (of [0.05, 0.1, 0.15, 0.2]) for a more comprehensive evaluation.

Generally, we implement our method upon the codebase of CapeFormer [45], which is built based on PyTorch [39] and MMPose [11]. Specifically, we use ResNet-50 [18] pre-trained on ImageNet [13] as our backbone, consisting with prior works [8, 45, 52]. For easy process, we use the same backbone to extract multi-scale feature maps from query and support images as in [45], and squeeze them to 256 channels using 1×1 convolutions, *i.e.*, $D = 256$. Our data configuration also follows previous works [8, 45, 52], *i.e.*, cropping the target object according to its GT bounding box and resizing the image patch to 256×256 . In the training stage, our data augmentations include random scaling and random rotation. We use Adam [25] optimizer to learn our model for 200 epochs using batch size 16, learning rate $1e^{-5}$. Our experiment environment is builed on Ubuntu 20.04 system with 64 GB Intel 9700K CPU and 2 NVIDIA 4090 GPUs.

4.2. Quantitative Comparison with Prior Works

4.2.1 Comparable Baselines. We follow previous works [8, 30, 45] and select state-of-the-art (SOTA) baselines for comparison. Considering the utilizing of link annotations, we categorize the selected baselines into following two groups. **(1) Point-based group.** POMNet [52] employs a keypoint matching framework to retrieve target keypoints on query images. CapeFormer [45] further refines the

matched keypoints with a novel transformer model. MetaPoint [8] first estimates potential keypoints, and then selects and refines them to target keypoints. ESCAPE [37] learns a prior over the features of keypoints, and then adapts them to target keypoints. SCAPE [30] focuses on learning high-quality attention to boost the performance of CAPE. **(2) Pose-based group.** GraphCape [19] treats the pose as a graph and uses a graph-based feed-forward network to extract geometrical features from keypoint feature vectors. For a comprehensive comparison, we firstly copy the already reported metrics and then reproduce the missing metrics by released weights or codes. We uniformly employ ResNet-50 [18] as the backbone network for fairness.

4.2.2 Result Analysis. All mPCK results in 1-shot setting and 5-shot setting are summarized in Tab. 1. As a pose-based method, GraphCape [19] achieves favourable performances against early work [52] by using graph network over keypoint feature vectors, and our method further outperforms dramatically (*e.g.*, +5.25 mPCK% on Split1) by using deformable attention to mine structure-aware features from both support and query images. Although MetaPoint [8] also employ deformable attention to mine fine-grained features from query image, our framework outperforms (*e.g.*, +1.61 mPCK% on Split1) by recurrently mining FGSA features from both support and query images. Overall, our model achieves the optimal performances against all SOTA baselines (*i.e.*, +1.06 mPCK% in average on Split1), demonstrating the effectiveness of our method.

4.2.3 More Detailed Comparison. In Tab. 2, we summarize the detailed PCK results of 0.05, 0.1, 0.15 and 0.2 thresholds on Split-1 in 1-shot setting. On the one hand, we could see that the threshold values significantly influence the performance gaps, *e.g.*, SCAPE [30] outperforms POMNet [52] +7.4% on PCK@0.2 while outperforms +9.7% on PCK@0.05, indicating that the performance measured by coarse threshold may be about saturated. On the other hand, our method outperforms the most competitive baseline (*i.e.*, SCAPE [30]) by a large margin on fine threshold (+3.2% on PCK@0.05), and also achieves dramatical improvement

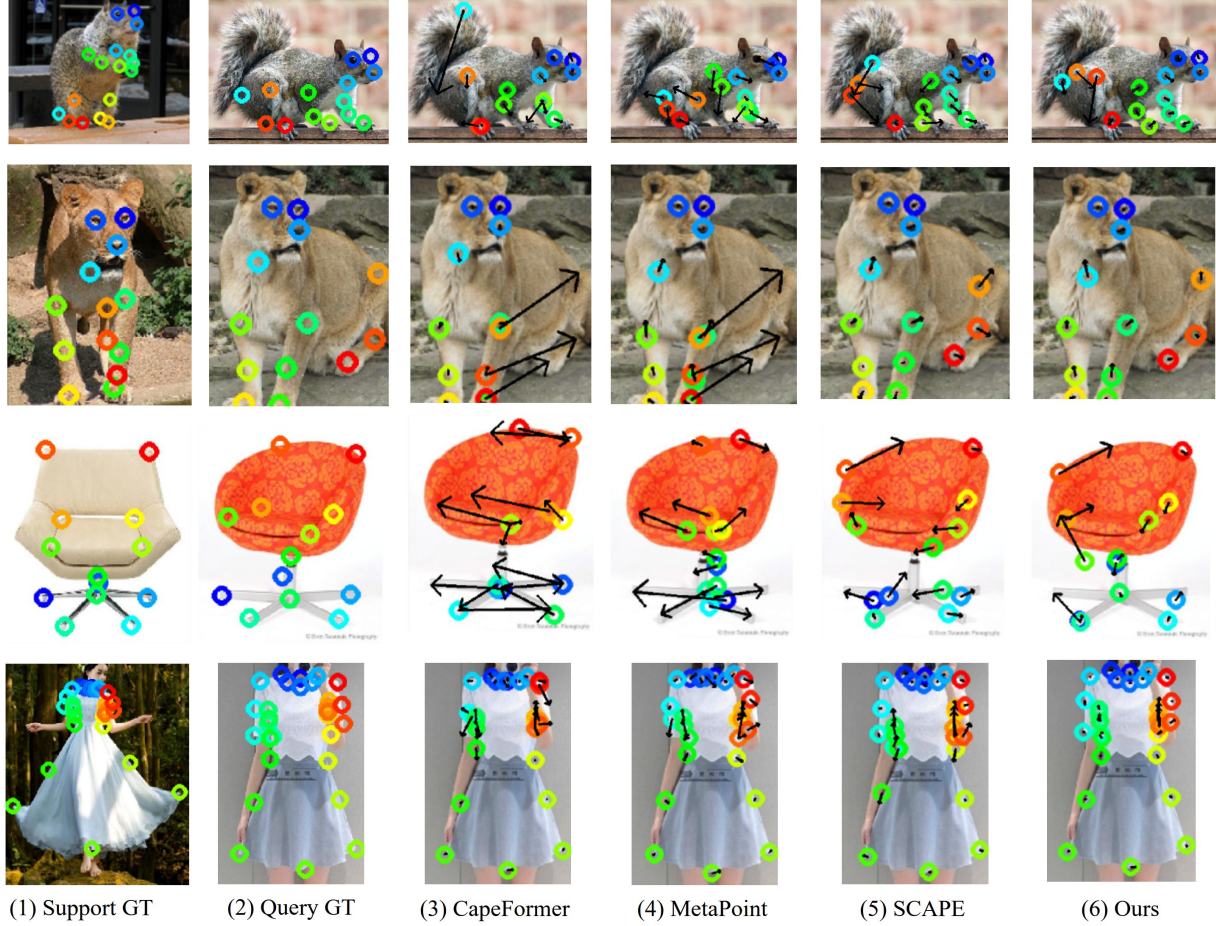


Figure 4. Qualitative comparison. The left two columns show the GT support and target keypoints on support and query images. The right four columns show the target keypoints on query images estimated by various methods. We employ graduated colors to denote the semantic classes of keypoints, and use small black arrows to indicate the deviations to GT target keypoints (*i.e.*, smaller arrows are better).

Table 2. The detailed PCK results using various thresholds (*i.e.*, 0.05, 0.1, 0.15 and 0.2) and their averaged mPCK on Split1.

Method	Th0.05	Th0.1	Th0.15	Th0.2	mPCK
POMNet[52]	44.39	68.87	79.39	84.23	69.22
CapeFormer[45]	51.03	75.17	84.87	89.45	75.13
ESCAPE [37]	48.24	72.25	82.30	86.89	72.42
MetaPoint[8]	55.08	77.12	85.81	90.43	77.11
GraphCape [19]	48.55	73.43	83.71	88.19	73.47
SCAPE [30]	54.09	77.34	87.02	91.67	77.53
Ours	57.30	78.48	87.28	91.82	78.72

on mPCK metric, demonstrating the effectiveness of recurrent FGSA feature mining.

4.3. Qualitative Comparison with Prior Works

In this section, we qualitatively compare our method with prior works. Specifically, we select representative methods to visualize the estimated keypoints on the test set of

Table 3. Results (mPCK) of our model in ablation study.

	$f_{miner-s}$	$\mathcal{P}_k$	f_{mixup}	$\mathcal{L}_{mixup}$	Split1
#1	-	-	-	-	69.82
#2	✓	-	-	-	73.18
#3	✓	✓	-	-	76.23
#4	✓	✓	✓	-	77.41
#5	✓	✓	✓	✓	78.72

dataset Split-1, including CapeFormer [45], MetaPoint [8], and SCAPE [30]. As shown in Fig. 4, our proposed method could estimate more precise keypoints for various object classes. *E.g.*, in the second row, the support keypoints are partially occluded, and our method can mine FGSA features and locate target keypoints more precisely. Besides, our method can better locate the dense keypoints for clothing in the last row, probably due to our keypoint mixup padding strategy. Overall, our method estimates finer and more accurate keypoints for various objects.

Table 4. Results (mPCK) of different versions of our model.

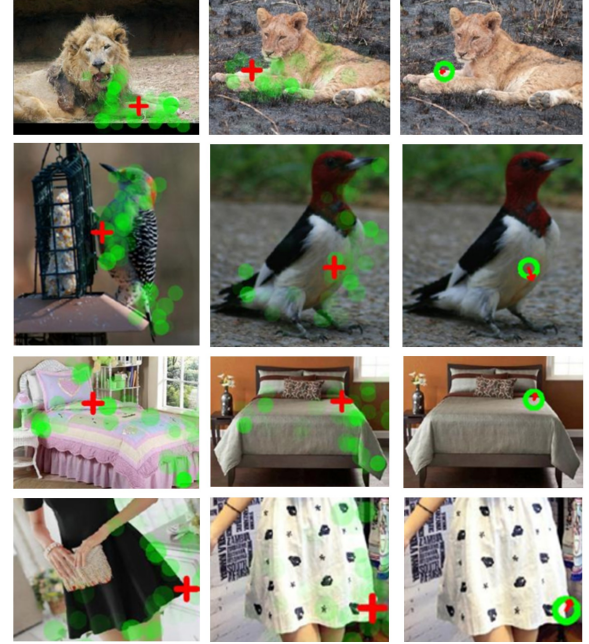
	Version	Split1	Split2	Split3	Split4	AVG
#1	Fledged	78.72	72.32	71.38	70.74	73.29
#2	S ³ Q ³	77.48	71.24	70.67	70.12	72.38
#3	AllLink	76.17	70.54	70.01	69.73	71.61
#4	NoneLink	75.23	70.12	69.53	69.05	70.98
#5	MixupTest	78.35	72.08	70.94	70.43	72.95
#6	ZeroTest	76.61	70.49	70.02	69.13	71.56

4.4. Method Analysis

4.5.1 Ablation Study. To investigate the performance contributions of our modules, we gradually append modules and record results in Tab. 3. Firstly, Row #1 represent our base model, which uses heatmap pooled features F_q^0 as support features and directly estimates target keypoints via $f_{miner-q}$ with conventional reference points and zero padding. In Row #2, we further recurrently mine features to update support features via $f_{miner-s}$, which indicates the effect of recurrent feature mining (*i.e.*, +3.35% mPCK). In Row #3, we set the reference points in $f_{miner-s}$ and $f_{miner-q}$ according to links to extract structure-aware features, which shows the gains of mining FGSA features from both support and query images (*i.e.*, +3.05% mPCK). Finally, we enable keypoint mixup padding in Row #4 and enable $\mathcal{L}_{mixup}$ in Row #5, which could provide denser semantics and filter improper supervisions for our model. Thus, all our modules are effective and complementary.

4.5.2 Configurations Analysis. Here we analyse different versions of our model and summarize the results in Tab. 4. Specifically, Row #1 shows the standard performance of our fledged model. Row #2 corresponds to a conventional pipeline, *i.e.*, using stacked layers to extract support features and then extract keypoint features with other stacked layers. By comparing Row #1 with #2, we can see the effects of our recurrent pipeline, which could extract targeted support features according to last keypoint features. In Row #3 and #4, we perturb the class-wise links to fully-connected (all-ones matrix) or only self-connected (identity matrix). Correspondingly, the performances degrade due to the corrupted structure, demonstrating our model can extract structure-aware features. In Row #5 with Row #6, we replace our default uniform padding in test stage to mixup padding or zero padding. Compared with Row #1, MixupTest slightly decreases the performance, may due to the randomness, and ZeroTest dramatically decreases due to sparser keypoints.

4.5.3 Qualitative Analysis. To intuitively understand how our model mines FGSA features, we visualize the attention points on support and query images in Fig. 5. We can see that the attention points (*i.e.*, sampling points) in deformable attention generally fit the structures of objects. *E.g.*, the attention points in the first row well fit the leg of lion in support and query images, and thus could provide



(1) Support Att. Points (2) Query Att. Points (3) Estimated Keypoints

Figure 5. Visualization for the attention on support image (col-1) and query (col-2) image, where the red cross indicates the reference point, and the green point indicates attention point with transparency as weight. The green circle in col-3 shows estimated keypoint and the red arrow shows the deviation to GT.

structural information for support features and keypoint features. Overall, the attention points could well capture the structures of various classes (*e.g.*, lion, bird, bed and clothing), and we can find similar phenomena in other cases.

5. Conclusion

In this paper, we have proposed a novel framework for CAPE by recurrent feature mining and keypoint mixup padding. Specifically, we have designed a feature mining module based on deformable attention, which could integrally mine fine-grained and structure-aware features. Besides, we have proposed a mixup padding strategy for richer supervision and denser semantics. We have conducted extensive experiments on MP-100 dataset, which demonstrates the effectiveness of our framework.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 62402201, 62271237 and U24A20220, in part by the Natural Science Foundation of Jiangxi Province of China under Grants 20242BAB26014 and 20242BAB21006, and in part by the Jiangxi Province Special Program for Cultivating Early-Career Young Scientific and Technological Talents under Grant 20244BCE52070.

References

- [1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, pages 3686–3693, 2014. 2
- [2] Yuxuan Bian, Xuan Ju, Jiangtong Li, Zhijian Xu, Dawei Cheng, and Qiang Xu. Multi-patch prediction: Adapting llms for time series representation learning. In *ICML*, 2024. 2
- [3] Bjorn Browatzki and Christian Wallraven. 3fabrec: Fast few-shot face alignment by reconstruction. In *CVPR*, pages 6110–6120, 2020. 2
- [4] Jinkun Cao, Hongyang Tang, Hao-Shu Fang, Xiaoyong Shen, Cewu Lu, and Yu-Wing Tai. Cross-domain adaptation for animal pose estimation. In *ICCV*, pages 9498–9507, 2019. 2
- [5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 3, 4, 5
- [6] Junjie Chen, Li Niu, Liu Liu, and Liqing Zhang. Weak-shot fine-grained classification via similarity transfer. In *NeurIPS*, pages 7306–7318, 2021. 2
- [7] Junjie Chen, Li Niu, Siyuan Zhou, Jianlou Si, Chen Qian, and Liqing Zhang. Weak-shot semantic segmentation via dual similarity transfer. In *NeurIPS*, pages 32525–32536, 2022. 2
- [8] Junjie Chen, Jiebin Yan, Yuming Fang, and Li Niu. Meta-point learning and refining for category-agnostic pose estimation. In *CVPR*, pages 23534–23543, 2024. 2, 4, 5, 6, 7
- [9] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, pages 7103–7112, 2018. 2
- [10] 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*, pages 5386–5395, 2020. 2
- [11] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. <https://github.com/open-mmlab/mmpose>, 2020. 6
- [12] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. 3
- [13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 6
- [14] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019. 3
- [15] Yuying Ge, Ruimao Zhang, and Ping Luo. Metacloth: Learning unseen tasks of dense fashion landmark detection from a few samples. *IEEE Transactions on Image Processing*, 31: 1120–1133, 2021. 2
- [16] Zigang Geng, Ke Sun, Bin Xiao, Zhaoxiang Zhang, and Jingdong Wang. Bottom-up human pose estimation via disentangled keypoint regression. In *CVPR*, pages 14676–14686, 2021. 2
- [17] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, pages 7297–7306, 2018. 2, 5
- [18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6
- [19] Or Hirschorn and Shai Avidan. Pose anything: A graph-based approach for category-agnostic pose estimation. In *ECCV*, 2024. 2, 6, 7
- [20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 3
- [21] Jian Hu, Hongya Tuo, Chao Wang, Lingfeng Qiao, Haowen Zhong, Junchi Yan, Zhongliang Jing, and Henry Leung. Discriminative partial domain adversarial network. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 632–648. Springer, 2020. 2
- [22] Jian Hu, Jiayi Lin, Shaogang Gong, and Weitong Cai. Relax image-specific prompt requirement in sam: A single generic prompt for segmenting camouflaged objects. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12511–12518, 2024. 3
- [23] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *NeurIPS*, 28, 2015. 3
- [24] Saumya Jetley, Nicholas A. Lord, Namhoon Lee, and Philip Torr. Learn to pay attention. In *ICLR*, 2018. 3
- [25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 6
- [26] Rollyn Labuguen, Jumpei Matsumoto, Salvador Blanco Negrete, Hiroshi Nishimaru, Hisao Nishijo, Masahiko Takada, Yasuhiro Go, Ken-ichi Inoue, and Tomohiro Shibata. Macaquepose: a novel “in the wild” macaque monkey pose dataset for markerless motion capture. *Frontiers in behavioral neuroscience*, 14:581154, 2021. 2
- [27] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *CVPR*, pages 10863–10872, 2019. 2
- [28] 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*, pages 11025–11034, 2021. 2
- [29] Jiangtong Li, Li Niu, and Liqing Zhang. Action-aware embedding enhancement for image-text retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1323–1331, 2022. 3
- [30] Yujia Liang, Zixuan Ye, Wenze Liu, and Hao Lu. Scape: A simple and strong category-agnostic pose estimator. In *ECCV*, 2024. 2, 6, 7
- [31] Changsheng Lu and Piotr Koniusz. Few-shot keypoint detection with uncertainty learning for unseen species. In *CVPR*, pages 19416–19426, 2022. 2
- [32] Changsheng Lu and Piotr Koniusz. Detect any keypoints: An efficient light-weight few-shot keypoint detector. In *AAAI*, pages 3882–3890, 2024. 3

- [33] Changsheng Lu, Hao Zhu, and Piotr Koniusz. From saliency to dino: Saliency-guided vision transformer for few-shot keypoint detection. *arXiv preprint arXiv:2304.03140*, 2023. 3
- [34] Changsheng Lu, Zheyuan Liu, and Piotr Koniusz. Openkd: Opening prompt diversity for zero- and few-shot keypoint detection. In *ECCV*, 2024. 2
- [35] Zhengxiong Luo, Zhicheng Wang, Yan Huang, Liang Wang, Tieniu Tan, and Erjin Zhou. Rethinking the heatmap regression for bottom-up human pose estimation. In *CVPR*, pages 13264–13273, 2021. 2
- [36] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, Zhibin Wang, and Anton van den Hengel. Poseur: Direct human pose regression with transformers. In *ECCV*, pages 72–88. Springer, 2022. 2
- [37] Khoi Duc Nguyen, Chen Li, and Gim Hee Lee. Escape: Encoding super-keypoints for category-agnostic pose estimation. In *CVPR*, pages 23491–23500, 2024. 2, 3, 6, 7
- [38] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In *ICCV*, pages 6951–6960, 2019. 2
- [39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019. 6
- [40] Duo Peng, Zhengbo Zhang, Ping Hu, Qihong Ke, David KY Yau, and Jun Liu. Harnessing text-to-image diffusion models for category-agnostic pose estimation. In *ECCV*, pages 342–360. Springer, 2025. 3
- [41] N Dinesh Reddy, Minh Vo, and Srinivasa G Narasimhan. Carfusion: Combining point tracking and part detection for dynamic 3d reconstruction of vehicles. In *CVPR*, pages 1906–1915, 2018. 2
- [42] Pengfei Ren, Yuanyuan Gao, Haifeng Sun, Qi Qi, Jingyu Wang, and Jianxin Liao. Dynamic support information mining for category-agnostic pose estimation. In *CVPR*, pages 1921–1930, 2024. 2, 3
- [43] Matan Rusanovsky, Or Hirschorn, and Shai Avidan. Capex: Category-agnostic pose estimation from textual point explanation. *arXiv preprint arXiv:2406.00384*, 2024. 2
- [44] Dahu Shi, Xing Wei, Liangqi Li, Ye Ren, and Wenming Tan. End-to-end multi-person pose estimation with transformers. In *CVPR*, pages 11069–11078, 2022. 2
- [45] 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*, pages 7308–7317, 2023. 2, 4, 5, 6, 7
- [46] 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*, pages 5452–5462, 2019. 2
- [47] Meiqi Sun, Zhonghan Zhao, Wenhao Chai, Hanjun Luo, Shidong Cao, Yanting Zhang, Jenq-Neng Hwang, and Gaoang Wang. Uniap: Towards universal animal perception in vision via few-shot learning. In *AAAI*, pages 5008–5016, 2024. 2
- [48] A Vaswani. Attention is all you need. *NeurIPS*, 2017. 3
- [49] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *CVPR*, pages 3156–3164, 2017. 3
- [50] Zhen Wei, Bingkun Liu, Weinong Wang, and Yu-Wing Tai. Few-shot model adaptation for customized facial landmark detection, segmentation, stylization and shadow removal. *arXiv preprint arXiv:2104.09457*, 2021. 2
- [51] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, pages 3–19, 2018. 3
- [52] 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*, pages 398–416, 2022. 1, 2, 5, 6, 7
- [53] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vit-pose: Simple vision transformer baselines for human pose estimation. *NeurIPS*, 35:38571–38584, 2022. 2
- [54] Jie Yang, Wang ZENG, Sheng Jin, Lumin Xu, Wentao Liu, Chen Qian, and Ruimao Zhang. KptLLM: Unveiling the power of large language model for keypoint comprehension. In *NeurIPS*, 2024. 2
- [55] Jie Yang, Ailing Zeng, Ruimao Zhang, and Lei Zhang. X-pose: Detecting any keypoints. In *ECCV*, pages 249–268, 2025. 2
- [56] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2878–2890, 2012. 6
- [57] 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. 2
- [58] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2, 3, 5
- [59] Hao Zhang, Kaipeng Zhang, Lumin Xu, Shenqi Lai, Wenqi Shao, Naning Zheng, Ping Luo, and Yu Qiao. Language-driven open-vocabulary keypoint detection for animal body and face. *arXiv preprint arXiv:2310.05056*, 2023. 2
- [60] Hao Zhang, Lumin Xu, Shenqi Lai, Wenqi Shao, Naning Zheng, Ping Luo, Yu Qiao, and Kaipeng Zhang. Open-vocabulary animal keypoint detection with semantic-feature matching. *International Journal of Computer Vision*, pages 1–18, 2024.
- [61] Xu Zhang, Wen Wang, Zhe Chen, Yufei Xu, Jing Zhang, and Dacheng Tao. Clamp: Prompt-based contrastive learning for connecting language and animal pose. In *CVPR*, pages 23272–23281, 2023. 2
- [62] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, pages 9308–9316, 2019. 3
- [63] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 2