

PoseImageNet: Pose Estimation for Extensive Classes Based on Rich Structure Prototypes

Junjie Chen, Hong Cao, Weixiang Tao, Yuming Fang*,
Jiebin Yan, and Yifan Zuo

School of Computer Science and Artificial Intelligence,
Jiangxi University of Finance and Economics, Nanchang, China

Abstract. It is intuitive for human vision system to perceive the object position, area and structure, which corresponds to the tasks of detection, segmentation, and pose estimation. Compared to the modern detectors or segmentors, existing pose estimators only cover specific classes (*e.g.*, human body) and restricted structures. Thus, recent works gather previous single-class pose datasets to construct multi-class pose datasets, but the richness of classes and structures are still limited. To this end, we label a new pose dataset upon ImageNet dataset, which contains extensive classes and structures. Due to the structure diversity in single class, we decompose into two sub-tasks based on structure prototypes: prototype labeling and pose annotating. In this way, we obtain 3500+ structure prototypes of 720 semantic classes. Although different prototypes may have diverse structures, there are some shared keypoints. Therefore, we propose an appropriate baseline with query-based model and matching-based supervision, where a query will learn the shared information across keypoints and focus on estimating the keypoints with similar characteristics. Extensive analyses and experiments indicate the quality of our dataset and the effectiveness of our method.

1 Introduction

Perceiving the object position, area and structure are fundamental abilities for human vision systems, and have extensive applications in our daily life. Correspondingly, the task of detection, segmentation, and pose estimation have received great attention in recent years in the field of computer vision. Modern detectors and segmentors [19, 45] could provide the positions and areas for extensive objects, while existing pose estimators only cover specific classes, *e.g.*, estimating human pose [1, 32] or vehicle pose [44, 53]. There are far more kinds of objects in the complex scenes of real-world applications, and the structure information could be supportive and pivotal for high-level tasks, *e.g.*, behavior analysis [52, 58], human-object interaction [17, 21], etc.

Considering the class restriction issue, recent works turn to the few-shot paradigm [4, 6, 31, 34, 35, 42, 51, 54, 61] and zero-shot paradigm [36, 46, 64, 65, 70, 71].

* Corresponding author.

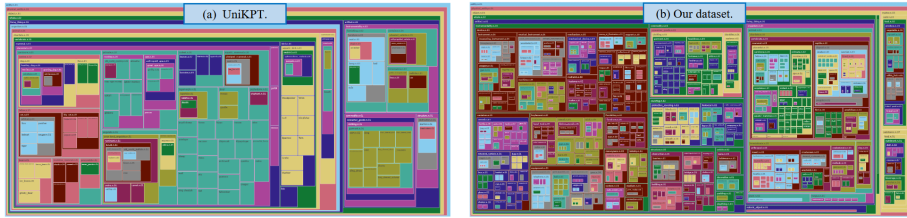


Fig. 1: The class hierarchy treemaps of UniKPT and our dataset. UniKPT [64] contains 4 super-classes, and ours contains 13 super-classes.

Although the above works have achieved great progress in extending the object scopes, there remains open problems in both benchmark dataset and technical method. On the one hand, existing multi-class datasets are the concatenation of previous single-class datasets. *E.g.*, UniKPT dataset [64] includes 13 datasets. As shown in Fig. 1, the class-wise richness of UniKPT dataset [64] is far less than ImageNet dataset [11] (about 16 years ago). On the other hand, existing few/zero-shot methods focus on learning transferable information. However, we may just need a standard fully-supervised model in some real-world scenarios, similar to the demands of fully-supervised classifiers, detectors, or segmentors. Therefore, a rich benchmark and an effective baseline are significant at present for extending the scope of pose estimation.

In this paper, we construct a rich benchmark and set an appropriate baseline for multi-class pose estimation. To construct the benchmark, we choose the ImageNet dataset [11] as our image source, which is also widely selected in related tasks, *e.g.*, part segmentation datasets [18] or sketchy datasets [49]. The first issue in annotating pose for such rich dataset is how to define the pose appropriately, because the object structures may obviously diverge in the same semantic class. *E.g.*, the phone class may contain foldable phones and bar phones, belonging to two structures and poses. Therefore, we propose to decompose into two sub-tasks based on structure prototypes: prototype labeling and pose annotating.

Specifically, for prototype labeling, we divide the images within the same class into several prototypes, based on the intuitive perception of structural similarity and imaginary transformations from images to prototypes. For pose annotating, we annotate the keypoints based on the insights that the pose could represent the major structures, and the images could be roughly warped and aligned to the their prototypes via the annotated keypoints. In this way, we obtain 3500+ structure prototypes of 720 semantic classes, which is dramatically richer than MP-100 [61] and UniKPT [64] datasets.

To set an appropriate baseline in the fully-supervised multi-class pose estimation, we propose a query-based model with matching-based loss. Previous fully-supervised pose estimators [33, 59] usually focus on single class and estimate in a fixed keypoint index, *i.e.*, the i -th channel estimates for the i -th keypoint. However, in the multi-class scenario, the i -th keypoint of some class may be similar to the j -th ($i \neq j$) of another class, instead of the i -th. *E.g.*, the eye keypoint

corresponds to index-3 of *bird* and index-7 of *dog*. Therefore, we design a query-based model and employ matching-based loss to adaptively match the queries to the keypoints. In this way, a query could learn the shared information across keypoints and focus on estimating the keypoints with similar characteristics.

We conduct extensive analyses and experiments to investigate the quality of our dataset and the effectiveness of our method. Our contributions can be summarized as follows: **(1)** We construct a significantly richer multi-class pose dataset based on ImageNet. We propose to firstly label the structure prototypes and then annotate the poses. **(2)** We propose a query-based model as an appropriate baseline, where the same query adaptively learn similar keypoints of different classes. **(3)** In-depth analyses and experiments demonstrate our dataset quality and method effectiveness.

2 Related Works

2D Pose Datasets Conventional 2D pose estimation usually focuses on single-class scenario, *e.g.*, human [1, 32], face [48], vehicles [44] and animals [3]. Recently, MP-100 [61] concatenates various single-class datasets (*e.g.*, DeepFashion2 [15], AP-10K [67], and MacaquePose [27]) to a multi-class pose dataset covering 100 classes. Later, UniKPT [64] further integrates 13 existing keypoint datasets (*e.g.*, COCO [32], 300W-Face [48], and Animal Kingdom [41]) to obtain a larger dataset. However, the class and structure richness of above datasets are limited, and thus we would like to expand to more extensive classes based on structure prototypes.

Single-Class Pose Estimation Previous pose estimation methods usually focused on keypoint localization for specific object class and fixed object structure, *i.e.*, with fixed keypoint number and keypoint types. Existing approaches can be divided into three main technical categories: regression-based methods [43, 56, 57] directly map image features to keypoint coordinates through an encoder structure, heatmap-based methods [8, 38, 69] generate spatial likelihood distributions of keypoints using an encoder-decoder architecture, and transformer-based methods [28, 29, 39, 50, 66, 68] leverage self-attention mechanisms to model global dependencies, achieving keypoint localization through a sequence-to-sequence paradigm. Although previous methods have achieved great success, the models are designed to learn a fixed structure, and thus lead to sub-optimal results for variable structures. In this paper, we construct a dataset and set an appropriate baseline to adaptively learn rich structures.

Few-Shot and Zero-Shot Pose Estimation To estimate the pose of multiple classes, recent works usually adopt few-shot [4–6, 26, 31, 35, 42, 51, 54, 61] and zero-shot [22, 23, 46, 64, 65, 70, 71] paradigms. Specifically, Lu *et al.* [34] propose the first few-shot keypoint detection, which estimates the keypoints of novel classes based on a small number of support samples. POMNet [61] locates keypoints through matching between query feature map and support keypoint features.

MetaPoint [6] learns transferable and shared meta-points from different categories and refines them to detect novel keypoints. In the zero-shot paradigm, Lu *et al.* [36] constructed a multi-modal prompt library and designed a multi-prompt contrastive learning framework for zero-shot pose estimation. Yang *et al.* [65] proposed to firstly recognize and then detect, which achieved cross-task keypoint detection. The above works have achieved great progress in learning transferable information and estimating the keypoints of novel classes. In this paper, we focus on multi-class pose estimation in the fully-supervised paradigm.

3 Dataset Construction

3.1 What is Pose?

This seemingly simple question is actually far more complex, which will define the characteristics of our final annotated poses. Our answer is that the visual pose could: (1) characterize the main structure of a single object image; (2) depict the main structural deformation across object images. In contrast, object sketches [12] can also characterize the main structure but fail to capture this deformation due to missing inter-image correspondences. Based on pose deformation, the visual poses could characterize the behaviors of animals or the placements of static objects.

Consequently, our major principles for dataset construction are also two-fold: (1) the poses could characterize the main structure analogously to object sketches [12], but with a distinct sparse representation. (2) the poses are deformable across a set of object images, which are represented by a prototype or canonical pose [6, 55]. In particular, the first principle is straightforward, similar to how human draw object sketches [12]. The second principle is also intuitive enough by comparing the major structures of object images, and we further formulate the deformation via Thin Plate Spline (TPS) transformation [2].

Specifically, TPS transformation is a classic non-rigid geometric transformation technique, based on interpolation over control points in the source image and their corresponding target points in the target image. As for our pose deformation, the keypoints could serve as control points for the TPS transformation (termed as keypoint-based TPS). In each column of Fig. 2, we could see how two objects are deformed to align via keypoint-based TPS. However, the objects in two different columns have different structures and thus cannot be well aligned, which should be divided into different prototypes. Extending for multiple images, if a set of object images could appropriately deform to each other via keypoint-based TPS, they should be in the same prototype, as shown in Fig. 3.

In the following, we firstly describe the image source of our dataset, and then introduce the specific annotating schemes. We also statistically analyse our dataset and compare it to prior pose, part, and sketch datasets.

3.2 Image Source

As a pioneer work on labeling pose dataset with structure prototypes, we require a large-scale of object images, which could be borrowed from classification

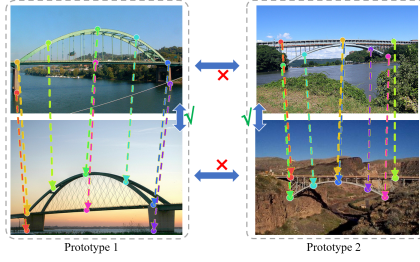


Fig. 2: The illustration for pose deformation. Four different *steel arch bridge* instances could be grouped into two prototypes based on deformation. In each prototype, we could intuitively deform one to the other using keypoints, but cannot well deform across prototypes.

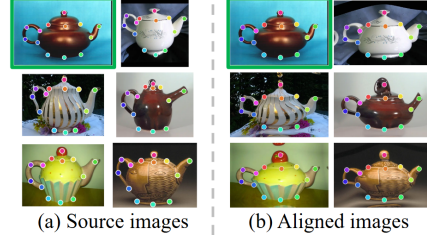


Fig. 3: Two global perspectives of samples in one prototype of *teapot* class. With the sample of green box as canonical pose, we warp the source images to aligned images using TPS transformation. It is intuitive to check the quality of keypoints.

dataset. ImageNet [11], containing about a large scale of training images with semantic class labels, is one of the most classic and fundamental benchmark datasets for various vision tasks. Therefore, using the images of ImageNet is an appropriate choice to bridge the gap for the research of pose estimation.

Annotating pose is relatively time-consuming, and thus it is quite expensive to provide pose annotations for all training images. As aforementioned, we focus on setting the dataset scheme and taxonomy, and control in a moderate scale comparable to related datasets [30], *i.e.*, up to 20 samples per prototype. We also include the pose samples from UniKPT by randomly selecting at most 200 samples per class, referring to the balanced number in [61]. Due to class overlapping, some fine-grained classes in ImageNet are merged and replaced to their coarse-grained classes in UniKPT, *e.g.*, *malamute*, *whippet*, *old English sheepdog* of ImageNet are updated with *dog* of UniKPT. Overall, we include 163 classes from UniKPT. Besides, we ignore 62 classes of ImageNet that have no obvious structure, *e.g.*, *seashore* and *cliff*. Finally, we have 720 classes totally. In the following, we introduce the details of labeling and annotating the images.

3.3 Labeling Prototype

To label some prototypes and assign each image with a prototype, we process the classes respectively and label the images sequentially. For each class, we maintain a set of structure prototypes $\mathcal{P} = \{\mathbf{x}_i\}$, where the set is initially empty and the i -th image in the set represent the i -th prototype. For each image, the annotators are required to determine whether to assign it to one of existing prototypes or create a new prototype using it, according to the principles introduced in Sec. 3.1. We also introduce how we check and correct the manual labels in Sec. 3.5. Specifically, in the labeling software interface, the image to be labeled is shown in the right sub-window, and the existing prototypes are illustrated in



Fig. 4: 3 structure prototypes and ignored samples of 2 classes. The postfix distinguishes prototypes in the same class.

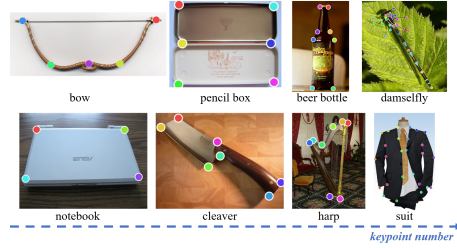


Fig. 5: The illustration of simple and complex object structures. The number of keypoints increases from left to right.

the left scroll. The annotators just click the illustrated prototypes or buttons to perform prototype labeling. Besides, if the image does not contain a clear structure (*e.g.*, due to the accumulation of small objects or heavy occlusions over large regions.), the annotators will label it as *ignore*. As shown in Fig. 4, 3 prototypes and 1 ignored sample are illustrated for *beer glass* and *chime* class.

3.4 Annotating Pose

After the above prototype labeling, all the samples are assigned with structure prototypes, and we annotate the poses for the prototypes respectively. In each prototype, the annotators firstly scan all the samples, and consider how to annotate major keypoints to align them with TPS transformation by using keypoints as control points. Considering that two difference instances inevitably cannot be aligned, we only requires the images to be aligned in common, general, or major structures. Similar to coarse-grained and fine-grained classification, we leave the fine-grained pose estimation to future works. Once the keypoints are determined for the first image in prototype, the subsequent pose annotating is more straightforward, because the transformation between the images in one prototype is intuitive. We employ the well-known tool LabelMe [47] for keypoint annotation, and the pipeline is conventional. As shown in Fig. 5, there are simple structures and complex structures in our prototypes, *e.g.*, 4 keypoints for *notebook* and 26 keypoints for *suit*.

3.5 Managing Quality

As the structure similarity or equivalence is apparent, checking the annotated poses from a global perspective is intuitive to find outliers. As shown in Fig. 3, we employ two kinds of global perspective to check the pose annotations of each prototype one by one. Firstly, we tile the source images depicting annotated poses to a grid of images, and ask expert annotators to check whether the annotated poses are in the same structure. Secondly, we also select a representative pose as the canonical pose, and align all the images belonging to the prototype to the

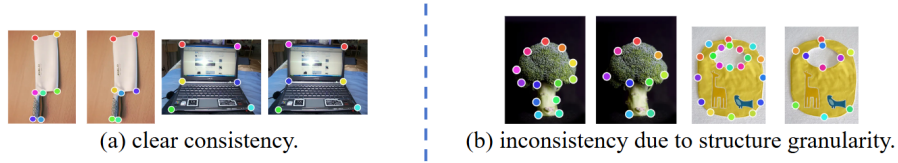


Fig. 6: Illustration for the consistent keypoint annotations and inconsistent keypoint annotations due to structure granularity.

canonical pose using keypoint-based TPS. From the two complementary global perspectives, the outliers (wrong annotations) are saliency, and thus the expert annotators could effectively identify them. For the identified outliers, the expert annotators will re-label the prototypes or poses.

Furthermore, we employ Fleiss’ Kappa [13] to show the consistency across annotators. For prototype labeling, we randomly select 1000 sample pairs with a balanced number of same-prototype and different-prototype pairs. We ask 5 annotators to determine whether each pair should be labeled into the same prototype, which is a binary classification task. As shown in Tab. 1, the result of Fleiss’ kappa is 82.7, which is larger than 80 and indicates the relatively high consistency in prototype labeling.

For pose annotating, we randomly select 20 prototypes and then randomly select 5 samples from each prototype. Since the keypoint numbers and orders of different annotators may be different, we have to match the keypoints and then compute PCK@0.2 to measure the consistency between two annotators. Specifically, we adopt two metrics: (1) PCK-All: the unmatched keypoints (due to extra keypoints) are regarded as errors; (2) PCK-Match: the unmatched keypoints are ignored. We compare every pairs of annotators and average to obtain the final metrics. As shown in Tab. 1, the results of PCK-All and PCK-Match are 71.6 and 98.2. In particular, PCK-All indicates the strict consistency, while PCK-Match allows finer-grained of structure depicting via extra keypoints, as shown in Fig. 6. Overall, the annotations exhibit satisfactory consistency.

3.6 Usability Analysis

We conduct a user study to verify whether our dataset can adequately depict object structures and deformations. Specifically, we draw 1000 random samples, and ask 5 users to rate “how well each pose depicts the main structure of the object” using a score in 0–5. We draw 1000 random sample pairs in the same prototype, and ask 5 users to rate “how well the left object could deform (move or be placed) into the pose of right object” using a score in 0–5. The averaged scores are summarized in Tab. 1, and thus our dataset could well reflect the two major principles of pose discussed in Sec. 3.1. Besides, we analyse the prototype diversity within each class via the variance of keypoint numbers. The averaged variances of classes (except for UniKPT) is recorded in the “Prototype Diversity”

Table 1: Analysis of annotator consistency, pose quality, and labeling time.

	Annotator Consistency			Pose Quality			Time	
	Prototype Labeling	Pose Annotation (PCK-All)	Pose Annotation (PCK-Match)	Structure Depicting	Deformation Depicting	Prototype Diversity	Prototype Labeling	Pose Annotation
Result	82.7	71.6	98.2	4.2	4.4	6.32	3s / img	2m / img

column of Tab. 1, where we could see that our classes generally contain diverse prototypes. Additionally, the averaged time consumption for labeling prototype and annotating pose are summarized in the last two columns of Tab. 1.

3.7 Comparison to Prior Related Datasets

In the domain of pose datasets, ours contains much more semantic classes and super-classes, *e.g.*, UniKPT [64] contains only *animal*, *vehicle*, *wear items*, and *furniture*. Prior pose datasets only cover one prototype for each semantic class, and we decompose classes into prototypes. UniKPT contains more pose annotations, by virtue of the concatenation of multiple prior single-class datasets. The part segmentation datasets (*i.e.*, PIN [7] and PIN++ [30]) are also related to our work, which all could be regarded as obtaining more fine-grained information on ImageNet. Specifically, single object in part segmentation datasets usually contains only several parts. *E.g.*, *left-hand* and *right-hand* are grouped into the same part *hand* in PIN++ [30]. In contrast we have more fine-grained semantics, *i.e.*, *left-hand* keypoint and *right-hand* keypoint.

Besides, sketch datasets [10, 14] may also depict the object structures, and thus related to our work to some extent. In particular, the object pose is a sparse representation, and contains semantic consistency. For example, sketches are difficult to depict the consistency or correlation across object images or video frames. Overall, our dataset are comparable to related datasets (*i.e.*, pose estimation, part segmentation, and sketches) and has particular characteristics.

4 Method

4.1 Setting of Fully-Supervised Multi-Class Pose Estimation

In our multi-class scenario, we set two basic tasks, *i.e.*, prototype classification and pose estimation. Specifically, in the training set, each data sample is given by input image $\mathbf{x} \in \mathbb{R}^{3 \times H \times W}$, prototype ID in one-hot vector $\mathbf{y}^* \in \mathbb{R}^N$, keypoint coordinates $\mathbf{P}^* \in \mathbb{R}^{K \times 2}$, and keypoint visibilities $\mathbf{V}^* \in \mathbb{R}^K$. Numerically, the coordinates are in range of $[0, 1]$ and the visibilities are binary. Because the keypoint types and numbers could vary across prototypes, we pad the poses into a unified dimension (*i.e.*, K) for inputting modern models, where the visibilities of padded keypoints are set as zero. In the inference stage, the models in our task should estimate the corresponding $\mathbf{y} \in \mathbb{R}^N$, $\mathbf{P} \in \mathbb{R}^{K \times 2}$, and $\mathbf{V} \in \mathbb{R}^K$ from input $\mathbf{x}$. For brevity, we employ $[\cdot]$ to denote the array index (*e.g.*, $\mathbf{P}[k]$ denotes the k -th keypoint), and adopt $[\cdot; \cdot]$ for the concatenation operator.

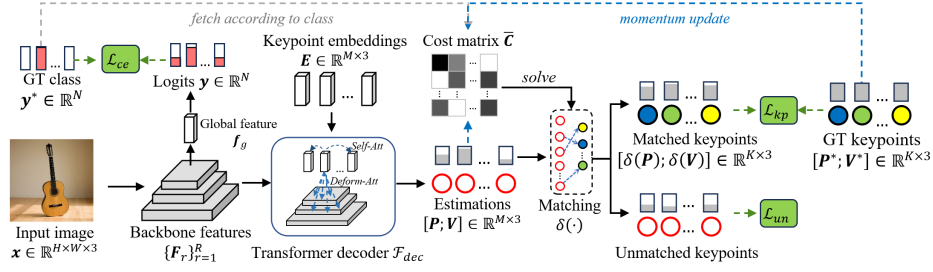


Fig. 7: The illustration of our model in the training stage. Firstly, our model classifies the structure prototype using global feature. After that, the shared keypoint embeddings extract keypoint features via a transformer decoder, and produce keypoint coordinates and visibilities. According to the GT class, the corresponding cost matrix is employed to solve the matching between estimations and targets. Finally, the keypoint regression and visibility classification are supervised based on matching. Besides, the cost matrix is updated via momentum.

4.2 Forward Pipeline

As shown in Fig. 7, given the input image $\mathbf{x}$, our method firstly extract the multi-scale feature maps $\{\mathbf{F}_r\}_{r=1}^3$ using a ResNet-50 backbone [20]. Conforming to conventional classification, we obtain a global feature vector $\mathbf{f}_g \in \mathbb{R}^D$ via global average pooling on $\mathbf{F}_3$ (the deepest layer), and employ a fully-connected layer to produce the structure classification logits, *i.e.*, $\mathbf{y} = \mathcal{F}_{cls}(\mathbf{f}_g) \in \mathbb{R}^N$, for N structure prototypes.

To estimate keypoints, we maintain $M \geq K$ learnable keypoint (query) embeddings $\mathbf{E} \in \mathbb{R}^{M \times D}$, which will learn shared information of keypoints across classes and prototypes. As keypoint estimating requires fine-grained features, we employ the deformable attention [72] based transformer to extract keypoint related features in image. We adopt the same architecture of deformable keypoint decoder in [6], which adopts deformable attention for keypoint-image interaction and adopts self-attention for keypoint-keypoint interaction. Specifically, we denote the deformable keypoint decoder as $\mathcal{F}_{dec}$, and employ keypoint embeddings to query on multi-scale backbone feature maps, which can be summarized as:

$$[\mathbf{P}; \mathbf{E}'] = \mathcal{F}_{dec}(\mathbf{E}, \{\mathbf{F}_r\}_{r=1}^3), \quad (1)$$

where $\mathbf{P} \in \mathbb{R}^{M \times 2}$ indicates the decoded keypoint coordinates, and $\mathbf{E}' \in \mathbb{R}^{M \times D}$ denotes the updated keypoint embeddings. Since the attention mechanism is widely applied and module design is not our contribution, the details of $\mathcal{F}_{dec}$ can be found in [6].

To estimate the visibilities of keypoints, we add a lightweight binary classifier over $\mathbf{E}'$, *i.e.*, $\mathbf{V} = \mathcal{F}_{cls_vis}(\mathbf{E}') \in \mathbb{R}^M$. Up to now, the estimated M keypoint coordinates $\mathbf{P} \in \mathbb{R}^{M \times 2}$ and visibilities $\mathbf{V} \in \mathbb{R}^M$ are in a sequence shared by all prototypes, and thus we need to determine (match) the corresponding indexes of the keypoints to the specific prototype in the input image.

4.3 Matching Estimations to Targets

Different prototypes have different types and numbers of keypoints, but modern models have to process padded and fixed outputs, *i.e.*, keypoints $\mathbf{P}^* \in \mathbb{R}^{K \times 2}$, and visibilities $\mathbf{V}^* \in \mathbb{R}^K$. Therefore, our designed model adaptively maps M estimations to K targets of each prototype respectively.

Specifically, we maintain one matching cost matrix for each prototype, *i.e.*, $\bar{\mathbf{C}} \in \mathbb{R}^{M \times K}$, while $\bar{\mathbf{C}}[m, k]$ stores the cost for matching the m -th estimated keypoint to the k -th target keypoint. The matching cost is initialized by zero matrix and updated via momentum accumulation in each training iteration, as:

$$\bar{\mathbf{C}} = (1 - \lambda)\bar{\mathbf{C}} + \lambda\mathbf{C}, \quad (2)$$

where λ is the momentum coefficient, and $\mathbf{C}$ is the immediate cost matrix computed using the targets and estimations of each training sample in training iteration. Similar to query-based models [6, 9], the matching cost is computed using loss functions, which will be formulated in Sec. 4.4. In particular, each entry in cost matrix is computed as:

$$\mathbf{C}[m, k] = \mathbf{V}[k] \cdot \left\| \mathbf{P}[m] - \mathbf{P}^*[k] \right\| + \alpha \mathcal{L}_{ce}(\mathbf{V}[m], \mathbf{V}^*[k]), \quad (3)$$

where the first term measures the cost between estimated and target coordinates, the second term measures the cost for visibility via binary cross-entropy, and α is the balance hyper-parameter. Based on the cost matrix $\bar{\mathbf{C}}$ of input sample, we solve the matching as in [9, 72], and obtain a mapping $\delta(\cdot)$, where the k -th target keypoint is matched to the $\delta(k)$ -th estimated keypoint. Because $M \geq K$, there are also $M - K$ unmatched estimations as in [9, 72].

4.4 Summary of Training and Inference

In each iteration of training stage, given the input image $\mathbf{x}$, GT prototype ID $\mathbf{y}^*$, and K keypoint targets $[\mathbf{P}^*; \mathbf{V}^*]$, our model forwards to predict prototype logits $\mathbf{y}$, and M keypoint estimations $[\mathbf{P}; \mathbf{V}]$, as aforementioned in Sec. 4.2. Then, we compute the immediate cost matrix $\mathbf{C}$ using GT targets and solve the matching mapping $\delta(\cdot)$ using the updated cost matrix $\bar{\mathbf{C}}$. After that, we employ the following learning objective to train the model:

$$\mathcal{L}_{full} = \mathcal{L}_{ce}(\mathbf{y}, \mathbf{y}^*) + \beta \cdot \sum_{k=1}^K \mathcal{L}_{kp}[k], \quad (4)$$

where $\mathcal{L}_{ce}$ is the cross-entropy loss for prototype classification, and β is the balance hyper-parameter. The $\mathcal{L}_{kp}[k]$ in the loss for learning the k -th target keypoint, and computed similar to Eqn. 3, which can be formulated as:

$$\mathcal{L}_{kp}[k] = \mathbf{V}^*[\delta(k)] \cdot \left\| \mathbf{P}[\delta(k)] - \mathbf{P}^*[k] \right\| + \alpha \mathcal{L}_{ce}(\mathbf{V}[\delta(k)], \mathbf{V}^*[k]), \quad (5)$$

where the first term supervises the coordinates of visible keypoint, and the second term supervises the visibility classification. Besides, there are $M - K$ unmatched estimations, and we supervise their estimated visibilities to be zeros (denoted as $\mathcal{L}_{un}$ in Fig. 7), which is trivial and omitted.

In inference, given the input image $\mathbf{x}$, our model forwards to predict prototype logits $\mathbf{y}$, and M keypoint estimations $[\mathbf{P}; \mathbf{V}]$. Then, we fetch the stored cost matrix $\tilde{\mathbf{C}}$ of predicted prototype class, and solve the matching mapping $\delta(\cdot)$. After that, our model outputs the logits $\mathbf{y}$ and matched K estimations, *i.e.*, $\delta(\mathbf{P}) \in \mathbb{R}^{K \times 2}$ and $\delta(\mathbf{V}) \in \mathbb{R}^K$.

5 Experiments

5.1 Implementation Details and Evaluation

We follow the implementation of related works [6, 51, 61], *e.g.*, image size, data augmentation, optimizer, etc. We employ $K = 100$, $M = 300$, $\lambda = 0.8$, $\alpha = 0.1$, and $\beta = 10$ by default. We random split the samples in each prototype into training and test samples with ratio 7 : 3. For prototype classification, we adopt the widely used accuracy metric, *i.e.*, Top-1 Acc, Top-3 Acc, and Top-5 Acc. For pose estimation, we choose PCK-based mAP, because the models should employ visibility to modulate the estimated structure. Specifically, we only count the GT visible keypoints, and the estimated keypoint corresponding to the k -th GT keypoint is considered to be valid if:

$$(\|\mathbf{P}[k] - \mathbf{P}^*[k]\| \leq T_p) \& (\mathbf{V}[k] \geq T_v), \quad (6)$$

where T_p is the distance threshold as PCK and T_v is the visibility/confidence threshold for AP. We mainly employ $T_p = 0.2$ and employ 9 visibility thresholds from 0.1 to 0.9. We employ the mean of all classes AP (*i.e.*, mAP@0.2) as our major metric.

5.2 Metric Computing

We choose PCK-based mAP metric, because the models should employ visibility to modulate the estimated structure. We firstly compute the AP value of each sample, and the aggregate them to a major metric. Given a distance threshold T_p , we compute AP value the i -th test sample with following processes. Firstly, we count the keypoint hit number N_{hit} as:

$$N_{\text{hit}} = \sum_k^K \mathbb{I}(\|\mathbf{P}[k] - \mathbf{P}^*[k]\| < T_p) \cdot \mathbb{I}(\mathbf{V}[k] > T_v) \cdot \mathbf{V}^*[k], \quad (7)$$

where T_v is the visibility/confidence threshold for computing AP, and $\mathbf{V}^*[k]$ is the GT visibility of the k -th keypoint. In this way, only those truly visible keypoints whose predicted position deviation by the model is lower than the threshold and whose visibility is higher than the threshold are considered correct.

Table 2: The accuracies (Top-1 Acc, Top-3 Acc, and Top-5 Acc ↑, %) in prototype classification task. The metric with “Ori” is the original accuracy of semantic classification on ImageNet.

ResNet	Ori Top1	Top-1	Top-3	Top-5	DenseNet	Ori Top1	Top-1	Top-3	Top-5
ResNet-18	72.1	59.8	86.8	90.6	DenseNet-121	74.9	62.8	89.1	92.6
ResNet-50	77.1	65.0	91.1	94.0	DenseNet-169	76.2	63.9	90.0	93.4
ResNet-101	78.2	66.6	91.9	95.1	DenseNet-201	77.4	65.6	91.6	94.7

Table 3: The performances (mAP@0.2↑, %) of different methods within various super-classes and all classes. We shorten some over-length super-class names for better table scale, including equipment, furniture, structure, plaything, wear items, and toiletry.

Method	<i>animal</i>	<i>device</i>	<i>vehicle</i>	<i>equip.</i>	<i>food</i>	<i>furni.</i>	<i>struc.</i>	<i>fungus</i>	<i>plant</i>	<i>tool</i>	<i>play.</i>	<i>wear.</i>	<i>toil.</i>	mAP@0.2
Sim.Base.	42.3	40.3	27.7	35.3	43.4	36.9	47.7	40.4	59.3	35.2	24.9	37.8	45.8	39.9
QueryPose	45.4	44.7	32.5	38.8	46.9	38.5	49.3	43.8	61.1	37.5	28.5	41.1	48.9	42.8
ViTPose	46.6	45.8	32.6	39.1	48.1	39.4	50.4	44.3	62.2	39.2	30.1	42.2	49.2	43.8
PCT	47.2	46.4	32.8	40.3	49.3	40.3	51.2	44.8	63.5	39.7	30.9	43.1	50.4	44.6
ED-Pose	48.1	47.1	33.2	40.9	50.4	41.5	52.1	46.1	63.9	41.1	31.2	43.9	51.2	45.4
GroupPose	49.2	49.7	35.9	43.5	52.6	43.9	55.1	48.9	65.6	43.2	33.1	46.2	53.1	47.7
NerPE	49.0	49.0	34.2	41.9	51.9	43.6	53.7	47.6	64.5	42.7	32.3	45.2	52.3	46.8
RTMO	49.9	49.9	36.8	43.2	52.9	44.1	55.9	49.3	66.3	43.9	33.4	46.4	54.1	48.2
Ours	52.0	58.6	54.5	50.8	63.8	56.3	70.4	70.4	74.1	55.3	49.4	62.3	63.8	57.0

And we compute the estimated keypoint number N_{est} and all GT keypoint number N_{all} as: $N_{\text{est}} = \sum_m^M \mathbb{I}(\mathbf{V}[m] > T_v)$, and $N_{\text{all}} = \sum_k^K \mathbf{V}^*[k]$. Correspondingly, the precise rate is $PR = \frac{N_{\text{hit}}}{N_{\text{est}}}$, and the recall rate is $RR = \frac{N_{\text{hit}}}{N_{\text{all}}}$. By iterating T_v from 0.1 to 0.9, we compute the AP under T_p for the i -th sample. We firstly average the AP results over samples, and then obtain the class-wise AP. Finally, we compute the mean of all classes to obtain final mAP under the distance threshold T_p . Besides, we could compute the mean of classes belonging to one super-class to obtain the mAP of super-class.

5.3 Structure Prototype Classification

The conventional classification mostly focuses on semantic classification, and we explore the structure classification of representative classifiers, including various scales of ResNet [20] and DenseNet [25]. As shown in Tab. 2, the classifiers achieve satisfactory performances, *e.g.*, 65.0 Top-1 Acc for ResNet-50. The performances of structure classification is relatively lower than semantic classification, may due to more class number and more fine-grained class cues. Besides, on Top-5 Acc, the classifiers could generally predict precise results, *e.g.*, 94.7 for DenseNet-201.

5.4 Extensive-Class Pose Estimation

We select representative pose estimation methods and adapt them to our baselines, including SimBase [59], QueryPose [60], ViTPose [62], PCT [16], ED-Pose [63], GroupPose [33], NerPE [24], and RTMO [37]. Specifically, for the

Table 4: Ablation study and fine metrics of our method.

Method	mAP@0.05	mAP@0.10	mAP@0.15	mAP@0.2	AVG
Sim.Base. [59]	2.6	12.8	26.4	39.9	20.4
RTMO [37]	17.5	36.7	43.5	48.2	36.5
Full-Fledged	27.2	43.3	51.8	57.0	44.8
Fix Matching	8.0	17.8	26.8	35.3	22.0
Using GT Class	38.6	61.9	73.3	79.5	63.3

heatmap-based methods, we employ the maximum value of each heatmap as the visibility. For the regression-based methods, we augment the one dimension to regress the visibility of each keypoint. We summarize the mAP@0.2 averaged within each super-class and the major mAP@0.2 averaged for all classes in Tab. 3.

From Tab. 3, we could see that our method achieves the optimal performance against various baselines, *e.g.*, +8.8 mAP@0.2 against RTMO [37]. For each super-class, our method also outperforms baselines dramatically, *e.g.*, +16.1 mAP@0.2 against GroupPose [33] for *wear items* super-class. The significant performance gains demonstrate the effectiveness of adaptively matching queries to learn shared information across keypoints.

Besides, we select the most competitive baseline (*i.e.*, RTMO [37]), to qualitative investigate the estimated poses. As shown in Fig. 8, our method could estimate object poses better depicting various structures. Specifically, in the 4-rd row of challenging *acoustic guitar* class, our estimated pose could generally represent the long strip shape of head and the curved shape of body. Overall, our method have the optimal quantitative and qualitative performances, indicating the effectiveness of our query-based model and matching-based supervision.

5.5 Ablation Study

In this section, we explore the impact of specific designs and report fine metrics of our method. Beyond the coarse metric mAP@0.2, we also report fine metrics (*i.e.*, mAP@0.05, mAP@0.10, and mAP@0.15) of the representative baseline and various versions of our method in Tab. 4. Specifically, *Full-Fledged* means the standard version of our method. *Fix Matching* means we replace the adaptive bi-partial matching with a fixed 1-to-1, 2-to-2, ..., K-to-K mapping. *Using GT Class* means we fetch the cost matrix according to GT class label in the test stage, instead of the original estimated class label.

As shown in Tab. 4, we could see that our full-fledged method consistently and significantly outperforms baseline on various grains of metric. However, with fix matching, our performances dramatically degrade, *e.g.*, -22.8% mAP in average, indicating the effectiveness of using matched query to learn shared information across keypoints. Using GT class labels, our performances significantly promote, *e.g.*, +18.5% mAP in average, showing the performance loss caused by wrong keypoint order. Overall, our method is an effective baseline for extensive class pose estimation, and we will further improve in future works.

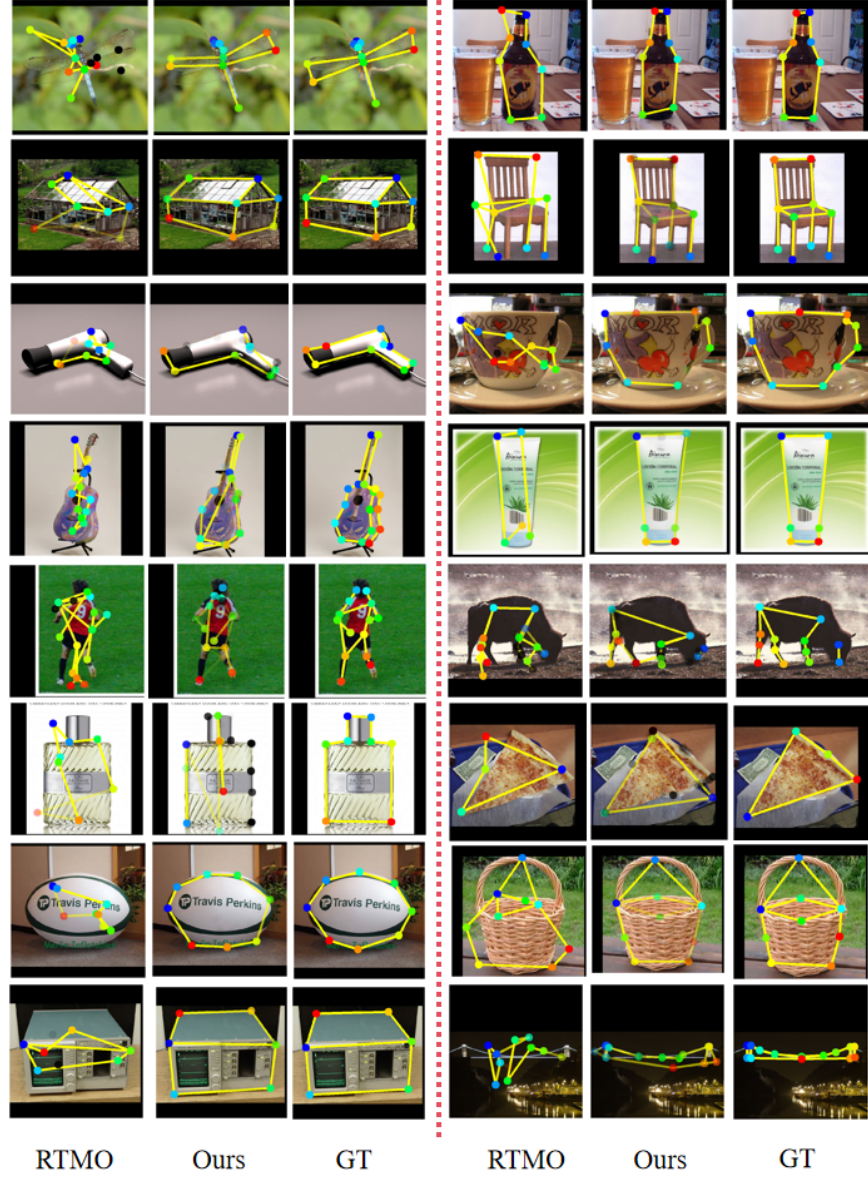


Fig. 8: The qualitative comparison for RTMO [37], ours, and GT keypoints. For clear visualization, we set various colors for GT visible keypoint and set black color for padded keypoint for each prototype. The keypoint transparency show the keypoint visibility.

5.6 Discussion, Limitation and Future Works

As a pioneer work for extensive-class pose estimation, our work inevitably contains various limitations and remains large scopes for further progress. We discuss the limitations from both the task setting and method design perspectives.

For task setting, we decompose one semantic class to multiple structure prototypes, but do not merge or assemble the prototypes according to structures, *e.g.*, *basketballs* and *baseballs* may be regarded as objects of the same structural prototype. Therefore, existing structure prototype classification actually classifies different prototypes under the semantic classes, instead of true structural characteristics (which may be semantically independent). Considering that the semantic classes have hierarchy or different levels of semantic granularities (*e.g.*, WordNet [40] or ImageNet [11]), we conjecture that structure prototypes may also have hierarchy, *e.g.*, some simple structures may be assembled to complex structures. However, it is not intuitive to hierarchically merge all prototypes at present, and we will explore this further in future work.

For method design, we employ prototype classification to fetch the cost matrix for matching, and thus the misclassification will lead to mismatching and incorrect keypoint orders. As discussed in Sec. 3.1, visual poses should depict object structures and be deformable across a set of objects. Correspondingly, a sample’s prototype estimation identifies the object image set it can deform to. Therefore, under misclassification, our method fails to identify the correct deformable set and produce the right keypoint order. However, it may still depict object structures well, since the keypoint coordinates are not affected. We will exploit more structure-discriminative cues to improve the performance of prototype classification task.

6 Conclusion

In this paper, we focus on estimating pose for extensive classes. To support this task, we have labeled and constructed a new pose dataset upon ImageNet dataset, containing extensive classes and structure prototypes. To learn the variable structures of extensive classes, we have set a baseline with query-based model and matching-based supervision. Extensive analyses and experiments have demonstrated the quality of our dataset and the effectiveness of our method.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants U24A20220, 62132006, 62402201, 62271237, 62461028, in part by the Natural Science Foundation of Jiangxi Province of China under Grants 20252BAC230003, 20242BAB26014, 20252BAC240197, in part by the China Postdoctoral Science Foundation under Grant 2025M771495, and in part by the Early-Career Young Scientists and Technologists Project of Jiangxi Province under Grant 20244BCE52070.

References

1. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2d human pose estimation: New benchmark and state of the art analysis. In: *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*. pp. 3686–3693 (2014)
2. Bookstein, F.L.: Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence* **11**(6), 567–585 (2002)
3. Cao, J., Tang, H., Fang, H.S., Shen, X., Lu, C., Tai, Y.W.: Cross-domain adaptation for animal pose estimation. In: *ICCV*. pp. 9498–9507 (2019)
4. Chen, J., Chen, W., Zuo, Y., Fang, Y.: Recurrent feature mining and keypoint mixup padding for category-agnostic pose estimation. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 22035–22044 (2025)
5. Chen, J., Luo, Z., Liu, Z., Jiang, W., Niu, L., Fang, Y.: Weak-shot keypoint estimation via keyness and correspondence transfer. *Advances in Neural Information Processing Systems* **38**, 75279–75299 (2026)
6. Chen, J., Yan, J., Fang, Y., Niu, L.: Meta-point learning and refining for category-agnostic pose estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 23534–23543 (2024)
7. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.: Detect what you can: Detecting and representing objects using holistic models and body parts. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 1971–1978 (2014)
8. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: *CVPR*. pp. 7103–7112 (2018)
9. Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. *Advances in neural information processing systems* **34**, 17864–17875 (2021)
10. Chowdhury, P.N., Sain, A., Bhunia, A.K., Xiang, T., Gryaditskaya, Y., Song, Y.Z.: Fs-coco: Towards understanding of freehand sketches of common objects in context. In: *European conference on computer vision*. pp. 253–270. Springer (2022)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: *2009 IEEE conference on computer vision and pattern recognition*. pp. 248–255. Ieee (2009)
12. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? *ACM Transactions on graphics (TOG)* **31**(4), 1–10 (2012)
13. Fleiss, J.L.: Measuring nominal scale agreement among many raters. *Psychological bulletin* **76**(5), 378 (1971)
14. Gao, C., Liu, Q., Xu, Q., Wang, L., Liu, J., Zou, C.: Sketchycoco: Image generation from freehand scene sketches. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 5174–5183 (2020)
15. Ge, Y., Zhang, R., Wang, X., Tang, X., Luo, P.: Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 5337–5345 (2019)
16. Geng, Z., Wang, C., Wei, Y., Liu, Z., Li, H., Hu, H.: Human pose as compositional tokens. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 660–671 (2023)
17. Gkioxari, G., Girshick, R., Dollár, P., He, K.: Detecting and recognizing human-object interactions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 8359–8367 (2018)

18. He, J., Yang, S., Yang, S., Kortylewski, A., Yuan, X., Chen, J.N., Liu, S., Yang, C., Yu, Q., Yuille, A.: Partimagenet: A large, high-quality dataset of parts. In: European Conference on Computer Vision. pp. 128–145. Springer (2022)
19. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
21. Hou, Z., Yu, B., Qiao, Y., Peng, X., Tao, D.: Detecting human-object interaction via fabricated compositional learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14646–14655 (2021)
22. Hu, J., Lin, J., Gong, S., Cai, W.: Relax image-specific prompt requirement in sam: A single generic prompt for segmenting camouflaged objects. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 12511–12518 (2024)
23. Hu, J., Lin, J., Yan, J., Gong, S.: Leveraging hallucinations to reduce manual prompt dependency in promptable segmentation. *Advances in Neural Information Processing Systems* **37**, 107171–107197 (2024)
24. Hu, S., Sun, H., Wei, D., Sun, X., Wang, J.: Continuous heatmap regression for pose estimation via implicit neural representation. *Advances in Neural Information Processing Systems* **37**, 102036–102055 (2024)
25. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
26. Jiang, W., Luo, R., Luo, Z., Zhao, X., Chen, J., Fang, Y.: Pointly-supervised weak-shot semantic segmentation via dual mapping transfer. In: 36th British Machine Vision Conference 2025, BMVC 2025, Sheffield, UK, November 24–27, 2025. BMVA (2025)
27. Labuguen, R., Matsumoto, J., Negrete, S.B., Nishimaru, H., Nishijo, H., Takada, M., Go, Y., Inoue, K.I., Shibata, T.: Macaquepose: a novel “in the wild” macaque monkey pose dataset for markerless motion capture. *Frontiers in behavioral neuroscience* **14**, 581154 (2021)
28. Li, J., Wang, W., Chen, J., Niu, L., Si, J., Qian, C., Zhang, L.: Video semantic segmentation via sparse temporal transformer. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 59–68 (2021)
29. Li, J., Zhou, Z., Zhang, J., Cheng, D., Jiang, C.: Hftcrnet: Hierarchical fusion transformer for interbank credit rating and risk assessment. *IEEE Transactions on Neural Networks and Learning Systems* pp. 1–15 (2024). <https://doi.org/10.1109/TNNLS.2024.3475484>
30. Li, X., Liu, Y., Dong, N., Qin, S., Hu, X.: Partimagenet++ dataset: Scaling up part-based models for robust recognition. In: European Conference on Computer Vision. pp. 396–414. Springer (2024)
31. Liang, Y., Ye, Z., Liu, W., Lu, H.: Scape: A simple and strong category-agnostic pose estimator. In: ECCV. pp. 342–360 (2024)
32. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
33. Liu, H., Chen, Q., Tan, Z., Liu, J.J., Wang, J., Su, X., Li, X., Yao, K., Han, J., Ding, E., et al.: Group pose: A simple baseline for end-to-end multi-person pose estimation. In: ICCV. pp. 15029–15038 (2023)
34. Lu, C., Koniusz, P.: Few-shot keypoint detection with uncertainty learning for unseen species. In: CVPR. pp. 19416–19426 (2022)

35. Lu, C., Koniusz, P.: Detect any keypoints: An efficient light-weight few-shot keypoint detector. In: AAAI. pp. 3882–3890 (2024)
36. Lu, C., Liu, Z., Koniusz, P.: Openkd: Opening prompt diversity for zero-and few-shot keypoint detection. arXiv preprint arXiv:2409.19899 (2024)
37. Lu, P., Jiang, T., Li, Y., Li, X., Chen, K., Yang, W.: Rtmo: Towards high-performance one-stage real-time multi-person pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1491–1500 (2024)
38. Luo, Z., Wang, Z., Huang, Y., Wang, L., Tan, T., Zhou, E.: Rethinking the heatmap regression for bottom-up human pose estimation. In: CVPR. pp. 13264–13273 (2021)
39. Mao, W., Ge, Y., Shen, C., Tian, Z., Wang, X., Wang, Z., den Hengel, A.v.: Poseur: Direct human pose regression with transformers. In: ECCV. pp. 72–88. Springer (2022)
40. Miller, G.A.: Wordnet: a lexical database for english. *Communications of the ACM* **38**(11), 39–41 (1995)
41. Ng, X.L., Ong, K.E., Zheng, Q., Ni, Y., Yeo, S.Y., Liu, J.: Animal kingdom: A large and diverse dataset for animal behavior understanding. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19023–19034 (2022)
42. Nguyen, K.D., Li, C., Lee, G.H.: Escape: Encoding super-keypoints for category-agnostic pose estimation. In: CVPR. pp. 23491–23500 (2024)
43. Nie, X., Feng, J., Zhang, J., Yan, S.: Single-stage multi-person pose machines. In: ICCV. pp. 6951–6960 (2019)
44. Reddy, N.D., Vo, M., Narasimhan, S.G.: Carfusion: Combining point tracking and part detection for dynamic 3d reconstruction of vehicles. In: CVPR. pp. 1906–1915 (2018)
45. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* **28** (2015)
46. Rusanovsky, M., Hirschorn, O., Avidan, S.: Capex: Category-agnostic pose estimation from textual point explanation. arXiv preprint arXiv:2406.00384 (2024)
47. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. *International journal of computer vision* **77**(1), 157–173 (2008)
48. Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: The first facial landmark localization challenge. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 397–403 (2013)
49. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: learning to retrieve badly drawn bunnies. *Acm Transactions on Graphics (TOG)* **35**(4), 1–12 (2016)
50. Shi, D., Wei, X., Li, L., Ren, Y., Tan, W.: End-to-end multi-person pose estimation with transformers. In: CVPR. pp. 11069–11078 (2022)
51. Shi, M., Huang, Z., Ma, X., Hu, X., Cao, Z.: Matching is not enough: A two-stage framework for category-agnostic pose estimation. In: CVPR. pp. 7308–7317 (2023)
52. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems* **27** (2014)
53. Song, X., Wang, P., Zhou, D., Zhu, R., Guan, C., Dai, Y., Su, H., Li, H., Yang, R.: Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In: CVPR. pp. 5452–5462 (2019)

54. Sun, M., Zhao, Z., Chai, W., Luo, H., Cao, S., Zhang, Y., Hwang, J.N., Wang, G.: Uniap: Towards universal animal perception in vision via few-shot learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 38, pp. 5008–5016 (2024)
55. Sun, W., Tagliasacchi, A., Deng, B., Sabour, S., Yazdani, S., Hinton, G.E., Yi, K.M.: Canonical capsules: Self-supervised capsules in canonical pose. *Advances in Neural information processing systems* **34**, 24993–25005 (2021)
56. Sun, X., Shang, J., Liang, S., Wei, Y.: Compositional human pose regression. In: *ICCV*. pp. 2602–2611 (2017)
57. Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. In: *CVPR*. pp. 1653–1660 (2014)
58. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: *Proceedings of the IEEE international conference on computer vision*. pp. 4489–4497 (2015)
59. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: *ECCV*. pp. 466–481 (2018)
60. Xiao, Y., Su, K., Wang, X., Yu, D., Jin, L., He, M., Yuan, Z.: Querypose: Sparse multi-person pose regression via spatial-aware part-level query. *Advances in Neural Information Processing Systems* **35**, 12464–12477 (2022)
61. Xu, L., Jin, S., Zeng, W., Liu, W., Qian, C., Ouyang, W., Luo, P., Wang, X.: Pose for everything: Towards category-agnostic pose estimation. In: *European conference on computer vision*. pp. 398–416. Springer (2022)
62. Xu, Y., Zhang, J., Zhang, Q., Tao, D.: Vitpose: Simple vision transformer baselines for human pose estimation. *NeurIPS* **35**, 38571–38584 (2022)
63. Yang, J., Zeng, A., Liu, S., Li, F., Zhang, R., Zhang, L.: Explicit box detection unifies end-to-end multi-person pose estimation. *arXiv preprint arXiv:2302.01593* (2023)
64. Yang, J., Zeng, A., Zhang, R., Zhang, L.: X-pose: Detecting any keypoints. In: *European Conference on Computer Vision*. pp. 249–268. Springer (2024)
65. Yang, J., Zeng, W., Jin, S., Xu, L., Liu, W., Qian, C., Zhang, R.: Kptllm: Unveiling the power of large language model for keypoint comprehension. *arXiv preprint arXiv:2411.01846* (2024)
66. Yang, S., Quan, Z., Nie, M., Yang, W.: Transpose: Keypoint localization via transformer. In: *CVPR*. pp. 11802–11812 (2021)
67. Yu, H., Xu, Y., Zhang, J., Zhao, W., Guan, Z., Tao, D.: Ap-10k: A benchmark for animal pose estimation in the wild. *arXiv preprint arXiv:2108.12617* (2021)
68. Yuan, Y., Fu, R., Huang, L., Lin, W., Zhang, C., Chen, X., Wang, J.: Hrformer: High-resolution vision transformer for dense predict. *NeurIPS* **34**, 7281–7293 (2021)
69. Zhang, F., Zhu, X., Dai, H., Ye, M., Zhu, C.: Distribution-aware coordinate representation for human pose estimation. In: *CVPR*. pp. 7093–7102 (2020)
70. Zhang, H., Xu, L., Lai, S., Shao, W., Zheng, N., Luo, P., Qiao, Y., Zhang, K.: Open-vocabulary animal keypoint detection with semantic-feature matching. *International Journal of Computer Vision* **132**(12), 5741–5758 (2024)
71. Zhang, X., Wang, W., Chen, Z., Xu, Y., Zhang, J., Tao, D.: Clamp: Prompt-based contrastive learning for connecting language and animal pose. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 23272–23281 (2023)
72. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159* (2020)