ADAPTIVE ANCHOR FOR ROBUST KEYPOINT LOCALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing keypoint localization methods mostly select pre-defined points like image center as anchors, then infer keypoint locations referring to anchors. Pre-defined anchors are sensitive to occlusions and crowded scenes, leading to degraded robustness. This paper proposes to detect Adaptive Anchor (AdaAnchor) for keypoint localization. Instead of relying on pre-defined rules, AdaAnchor is adaptively selected by maximizing both the keypoint localization confidence and accuracy. This strategy leads to more robust keypoint localization even with the existence of occlusions and truncations. AdaAnchor can be flexibly integrated into different methods by replacing their anchor point selection strategies. Experiments show that it surpasses previous anchor selection methods on both single and multiple keypoint localization tasks. For instance, replacing the heatmap-anchor with AdaAnchor reduces the localization error of invisible keypoints by 6%, meanwhile improves the confidence by 41.7+% on COCO in single keypoint localization. This advantage sustains on multiple keypoint localization task, *e.g.*, AdaAnchor outperforms heatmap-anchor by 4.8% AP on bottom-up multi-person pose estimation.

1 INTRODUCTION

Keypoint localization is a fundamental task in computer vision. It has a wide range of applications in various fields, such as human pose estimation (Sun et al., 2019; Xiao et al., 2018; Cao et al., 2017; Li et al., 2021a) and facial landmark detection (Wu et al., 2018; Sagonas et al., 2013). Existing methods for keypoint localization can be divided into two categories: heatmap-based (Sun et al., 2019; Xiao et al., 2018) and regression-based (Li et al., 2021a; Toshev & Szegedy, 2014) methods, respectively. Heatmap-based methods use a heatmap to encode the likelihood of the target keypoint locations. Keypoints can be retrieved by selecting locations showing the highest probability. Regression-based methods directly regress coordinates of target keypoints. Heatmap-based methods show better performance, but commonly consume more computations and are more sensitive to occlusions. Regression-based methods enjoys better efficiency, but are usually inferior to heatmap-based methods in localization accuracy.

Heatmap-based and regression-based methods follow different pipelines, but they share a similar spirit in using anchor points, *i.e.*, first selecting an anchor as the reference, then inferring target keypoints based on this anchor. As shown in Fig. 1, heatmap-based methods usually select a point near the target location as the anchor, then regress offsets of coordinates (Papandreou et al., 2017). Some methods like (Xiao et al., 2018; Sun et al., 2019) select anchors from high-resolution feature maps, and directly adopt those anchors as keypoint localizations. Regression-based methods define the top-left or center point of the input image as anchors. As illustrated in Fig. 1, both heatmap-based and regression-based methods use pre-defined rules to locate anchors, hence show degraded robustness in complex scenarios. For instance, the target keypoint could be invisible due to occlusions or truncation, leading to the difficulty of anchor detection in heatmap-based methods. Regression-based methods are more robust to occlusion or truncation by simply choosing a fixed anchor. However, fixed anchors could be far away from target keypoints, making keypoint regression difficult. Fixed anchors are also not applicable to multiple keypoint localization tasks due to their limited capability in differentiating similar instances.

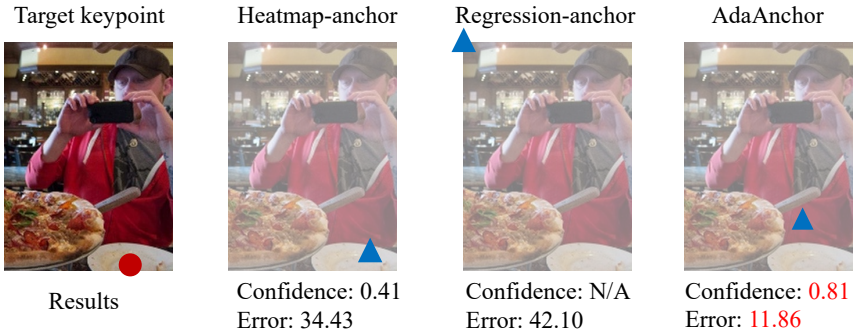

Figure 1: Illustration of selected anchors by heatmap, regression, and the AdaAnchor under the same setup. To detect the target keypoint on left hip (denoted by the red circle in the first image), heatmap selects anchor near the target location. Regression selects the top-left point as anchor. They suffer from occlusions and the difficult of long-range regression, respectively. The proposed method adaptively selects anchors, and produces more reliable results.

This paper aims to adaptively select anchors to facilitate the keypoint localization. We propose Adaptive Anchor (AdaAnchor) to handle occlusions, and to ensure more confident and accurate keypoint localization. For each visible keypoint, AdaAnchor selects its nearby points as the anchor to guarantee the regression quality. For the invisible keypoint, AdaAnchor aims to select an anchor from a related visible region to avoid low confident or missed detection. AdaAnchor is expected to be a plug-and-play component to replace pre-defined anchors in different keypoint localization algorithms.

AdaAnchor selects anchors from a candidate pool. The training stage assigns a score to each candidate to indicate its importance in anchor selection. Each score reflects the probability that a candidate can successfully infer target locations with high confidence. Candidates with large score in each image hence can be selected as anchors for each. We define the score as a combination of localization confidence and accuracy. The localization confidence is computed as the probability of this candidate to detect target keypoint. The localization accuracy is a regression score reflecting localization accuracy. We model the target location by a gaussian distribution to convert the estimated coordinates into scores. We further introduce a multiple anchors strategy that selects multiple candidates for one target, as well as a local refinement module to relieve the long-range location regression issue. Those strategies lead to better keypoint localization performance.

AdaAnchor is general and can be flexibly integrated into existing methods. We test it on single keypoint and multiple keypoint localization tasks, *i.e.*, single person pose estimation, multi-person pose estimation and facial landmark detection, respectively. Experiments show that our method consistently outperforms existing heatmap-based and regression-based methods on different dataset and backbones. For instance, replacing heatmap-anchor with AdaAnchor boosts the performance of SimplePose (Xiao et al., 2018) and HRNet (Sun et al., 2019), by 0.8% PCKh@0.1 on MPII (Toshev & Szegedy, 2014), 0.16 NME on WFLW (Wu et al., 2018), respectively. It also shows superior performance in localizing invisible keypoints. AdaAnchor reduces the localization error of invisible keypoints by 6%, meanwhile improves the confidence over 41.7% on COCO (Lin et al., 2014). The advantages of AdaAnchor are more substantial on multiple keypoint localization task, where confidence is equally important. AdaAnchor outperforms pre-defined anchors by 4.8% under Associative Embedding (Newell et al., 2017) on bottom-up multi-person pose estimation and 0.8% under DEKR (Geng et al., 2021) on single-stage multi-person pose estimation. These experiments demonstrate the superior performance of the proposed method.

2 Related Work

Regression-based Keypoint Localization. A straightforward way to locate keypoint is to directly predict keypoint coordinates, which is adopted by several classical methods (Toshev & Szegedy, 2014; Carreira et al., 2016). Due to the inferior performance of direct regression, several works have been proposed to improve the performance of regression-based methods. A line of work changes

the way of regression. Integral pose regression (Sun et al., 2018) leverage the soft-argmax operation to regress keypoint locations by integrating on a latent heatmap, which is proved to be superior over direct regression. Sampling-argmax (Li et al., 2021b) further improves soft-argmax by minimizing the error between samples drawing from a distribution with groundtruth, avoiding unconstrained probability map in previous method. Some work improves regression by proposing new loss function. RLE (Li et al., 2021a) changes the predefined gaussian or laplace distribution in commonly used regression loss with a learned distribution via normalizing flow, leading to better regression results. Recently, with the popular of transformer-based model, researchers also try to improve the performance of direct regression by proposing more powerful backbones, such as Tokenpose (Li et al., 2021c) and PETR (Shi et al., 2022).

Heatmap-based Keypoint Localization. Heatmap-based methods widely adopt probability map to locate keypoint position, which is introduced by Tompson et al. (2014). Heatmap simplifies the previous regression task into an easier classification task, therefore shows superior performance over regression-based methods and dominates the field of keypoint localization. Pioneer works (Xiao et al., 2018; Sun et al., 2019; Newell et al., 2016) design powerfull CNN models to estimate heatmaps for human pose estimation and facial landmark detection. However, due to the limitation of feature map size, heatmap has quantization error and performs badly on low resolution inputs. To solve this problem, some works (Newell et al., 2016; Zhang et al., 2020) propose post-processing operation to relieve this error. The others combine a offset branch to avoid quantization error. These methods usually adopt vector field to encode the keypoint location, which contains both confident score and offsets. PersonLab (Papandreou et al., 2018) proposes the short-range offset to encode the keypoint spatial location. PifPaf (Kreiss et al., 2019) further extend it to part intensity field and include the uncertainty of each keypoint. These methods improve the performance of heatmap. However, they still adopt the pre-defined anchor for localization and cannot deal with occlusion and truncation.

3 METHOD

3.1 OVERVIEW

Keypoint localization aims to detect the numerical coordinates of target keypoint in an image, *i.e.*,

$$\{\mathcal{P}_k\}_{k=1}^{n} = \text{locate}(\mathcal{I}), \tag{1}$$

where $\mathcal{P}_k$ is the position of k-th keypoint, in this paper we focus on 2D keypoint, namely $\mathcal{P}_k = (x_k, y_k)$. n represents the number of detected targets in image $\mathcal{I}$, which is equal to 1 in single keypoint localization task and large than 1 in multiple keypoint localization task, such as bottom-up multi-person pose estimation.

As state above, existing methods select anchors to locate keypoints. Specifically, given an image $\mathcal{I} \in \mathbb{R}^{3\times H\times W}$ and a backbone $\Phi(\cdot)$, *i.e.*, Convolutional Neural Network (CNN), with output stride t, we can define a set of candidate anchors $\mathbb{A}$ as,

$$\mathbb{A} = \{\mathcal{A}\} = \{(i,j)\}_{i=1\ldots h}^{j=1\ldots w} \tag{2}$$

where $\mathcal{A} = (i,j)$ denotes a candidate with corresponding feature $f = \mathcal{F}_{i,j}, \mathcal{F} = \Phi(\mathcal{I})$. $h = H/t, w = W/t$ is the size of output feature map.

To detect target keypoint $\mathcal{P} = (x,y)$, we first need to select an anchor $\mathcal{A} = (i,j)$ from $\mathbb{A}$, then force the model to output high confidence c on $\mathcal{A}$ and low confidence on others, the selection process can be conceptually denoted as,

$$\mathcal{A} = \text{Select}(\mathbb{A}, \mathcal{P}) \tag{3}$$

A regressor is applied to further regress the offset $(\delta x, \delta y)$ between anchor $\mathcal{A}$ and target $\mathcal{P}$. Therefore the final localization result is composed as $(c, i+\delta x, j+\delta y)$. We learn the model by adding constraint on the confidence and regressed results, the overall training objective is denoted by,

$$\mathcal{L} = \underbrace{\mathcal{L}_{\text{cof}}(c, c^*)}_{\text{all candidates}} + \lambda \underbrace{\mathcal{L}_{\text{reg}}((\delta x, \delta y), (\delta x^*, \delta y^*))}_{\text{selected anchors}}, \tag{4}$$

where λ is a hyperparameter to balance the importance of the confidence and regression parts. c^* is 1 for selected anchor and 0 for others. $\mathcal{L}_{\text{reg}}$ is optional when the feature map is large enough to

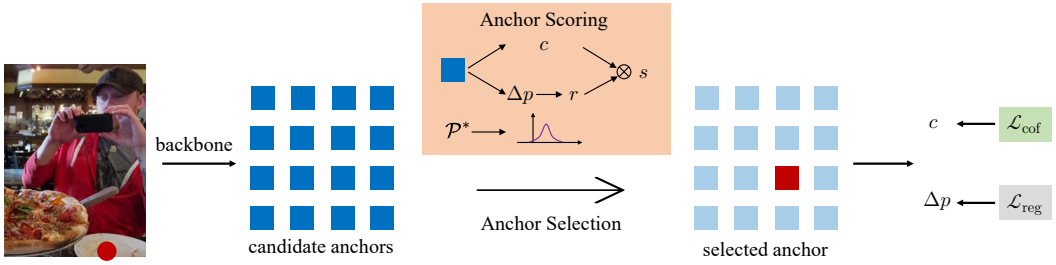

Figure 2: The training pipeline of the proposed AdaAnchor for keypoint localization. We use a backbone to extract feature map and get a set of candidate anchors. We assign a score to each candidate by considering both the confidence and accuracy of localization results, and select candidates with large score as anchors. Then we add constraint on the selected anchor to train model. During inference, we directly select the candidate with maximum confidence score as final result.

avoid quantization error. Following previous works (Zhou et al., 2019), we implement $\mathcal{L}_{\text{cof}}$ as Focal Loss and $\mathcal{L}_{\text{reg}}$ as L1 loss for their superior performance on classification and regression tasks.

During inference, we can simply select the candidate with large score as final anchor to obtain localization results. For single keypoint localization task, $\mathcal{A}$ can be simply selected as the candidate with maximum confidence. For multiple keypoint localization, a threshold γ is applied to filter out low confident candidates, and the remaining anchors are viewed as final results with $c \geq \gamma$.

It can be observed that, the anchor $\mathcal{A}$ plays an important role in localization, it can be viewed as the proxy of target keypoint. Therefore the choice of anchor is crucial for robust localization. Firstly, $\mathcal{A}$ should be easily selected with high confidence $c \geq \gamma$ to avoid missed localization, especially in occluded scenarios. Secondly, $\mathcal{A}$ should not be too far away from target because it is hard to accurately regress large offset $(\delta x, \delta y)$, which degrades the localization accuracy.

The pre-defined anchor selection of equation 3 in previous works cannot guarantee these requirements. Different from these, we are motivated to propose AdaAnchor that adaptively select proper anchor to detect target keypoint by maximizing both confidence and accuracy of final results, which is robust to occlusion and truncation. We replace equation 3 as our proposed method,

$$\mathcal{A} = \text{AdaAnchor}(\mathbb{A}, \mathcal{F}, \mathcal{P}) \tag{5}$$

When there are multiple targets in $\mathcal{I}$, we select multiple anchors $\{\mathcal{A}\}$ accordingly. Once the anchors are selected, we apply equation 4 to train the model. Note that our method keeps the same inference pipeline with previous methods, thus it is flexible and easy to apply to other methods without any test-time overhead. In following we will show how to select AdaAnchor for robust localization.

3.2 Adaptive Anchor Selection

The AdaAnchor is proposed to generate both high confident and accurate localization results in complex scenarios. Specifically, we first associate a score to each candidate to reflect their performance in final localization results. Then candidates with large score are selected as final anchors. To further improve the localization accuracy of AdaAnchor, we propose a multiple anchors strategy to select multiple candidates for each target, as well as a local refinement module to relieve the long-range location regression issue.

Anchor Scoring. As shown in Fig. 3.1, given a set of candidate anchors $\mathbb{A}$, we aim to assign each candidate with a score, indicating whether it should be selected or not. A candidate anchor with high score should produce both high confident and accurate localization results. Guided by these two criterions, we define the score as a combination of localization confidence and accuracy. The first part is a confidence score reflecting the probability of this candidate to detect target. The second part is a regression score reflecting the accuracy of localization.

Considering that we want to locate target keypoint $\mathcal{P}^* = (x^*, y^*)$ by given a set of candidates $\mathbb{A}$ and corresponding features $\mathcal{F}$. We first use two separate branches to estimate the detecting probability

$c_{i,j}$ and target keypoint offset $(\delta x_{i,j}, \delta y_{i,j})$ for each candidate, *i.e.*,

$$c_{i,j} = \text{CofHead}(\mathcal{F}_{i,j}), \tag{6}$$

$$(\delta x_{i,j}, \delta y_{i,j}) = \text{RegHead}(\mathcal{F}_{i,j}) \tag{7}$$

The estimated target position can be obtained as $(i + \delta x_{i,j}, j + \delta y_{i,j})$.

$c_{i,j}$ can be directly used as the first score to model the existence of target keypoint. To model the accuracy of localization, we need to translate the estimated $(i+\delta x_{i,j}, j+\delta y_{i,j})$ into a score. Inspired by the heatmap generation in previous methods (Xiao et al., 2018; Sun et al., 2019), we assume the target $\mathcal{P}^*$ follows an unnormalized gaussian distribution, *i.e.*,

$$p(x, y|x^*, y^*, \sigma) = \exp(-\frac{|x - x^*|^2 + |y - y^*|^2}{2\sigma^2}) \tag{8}$$

wherer σ is a hyperparameter that controls the variance of the distribution, we set it to 4 for all experiments. Therefore we can score the estimation $(i + \delta x_{i,j}, j + \delta y_{i,j})$ by equation 8 and get regression score $r_{i,j} = p(i + \delta x_{i,j}, j + \delta y_{i,j}|x^*, y^*, \sigma)$.

Once we get the confidence score $c_{i,j}$ and regression score $r_{i,j}$, the final score for each candidate can be obtained by fusing these two scores via multiplication,

$$s_{i,j} = c_{i,j} * r_{i,j}, \tag{9}$$

We also test other fusion strategy, *i.e.*, addition. We will evaluate their effectiveness in Sec. 4.1. The final score $s_{i,j}$ provides a guidance for the following anchor selection.

Anchor Selection. Given a set of candidates $\mathbb{A}$ with corresponding scores $\{s_{i,j}\}$. Our goal is to find the anchors to detect target keypoints. For single keypoint localization task, there only exists one target for each type of keypoint, therefore we can directly select the candidate with largest score as positive anchor, and the remaining to the negative, which can be denoted as,

$$\mathcal{A}^* = \arg\max_{\mathcal{A} \in \mathbb{A}}(\{s_{\mathcal{A}}\}) \tag{10}$$

For multiple keypoint localization task, more than one target keypoints are required to assign corresponding anchors. Suppose there are n targets $\{\mathcal{P}_k^*\}_{k=1}^n$ in $\mathcal{I}$, we assign each candidate with n scores $\{s_{i,j}^k\}_{k=1}^n$ by applying scoring process on each $\mathcal{P}_k^*$. Then we simply adopt equation 10 separately on each type of score $\{s^k\}$ to obtain corresponding anchor $\mathcal{A}_k^*$.

Multiple Anchors. We find that the above single anchor selection strategy cannot provide enough positive supervision to train model. Moreover, in our adaptive setting, the optimal anchor may not be unique, especially when the target is occluded or out of image. Therefore we propose to extend it to select top-k candidates as the final anchors, *i.e.*,

$$\{\mathcal{A}^*\} = \underset{\mathcal{A} \in \mathbb{A}}{\text{top-k}}(\{s_{\mathcal{A}}\}) \tag{11}$$

where k is a hyper-parameter for selection. In experiments we find that a properly large k can constantly improve the performance compared with single anchor selection strategy.

Local Refinement. The above anchor scoring and selection can already provide good results for keypoint localization. However, we find that there exists a contradiction between high confidence and high accuracy, especially when occlusion and truncation exist. For high confident localization, the model tends to select the visible part as anchor, which may be far away from target. And due to the long distance regression, it is hard to accurately regress large offset, small interference will introduce large detection error. To ensure high accuracy, the model tends to select anchors near the target, which results in low confident or missed detection.

To solve this problem, we introduce a local refinement module to relieve the long distance regression. The motivation is that, the estimated $(i+\delta x, j+\delta y)$ can provide a coarse but not precise localization result, and the local offset around the groundtruth can be easily estimated, which is also verified in previous work (Papandreou et al., 2017; Mao et al., 2021). So we can refine the coarse estimation $(i + \delta x, j + \delta y)$ in equation 6 with a local precise information. As shown in Fig. 3.2, we use two branch to estimate two offset maps, the first is the same as the anchor scoring module, which is

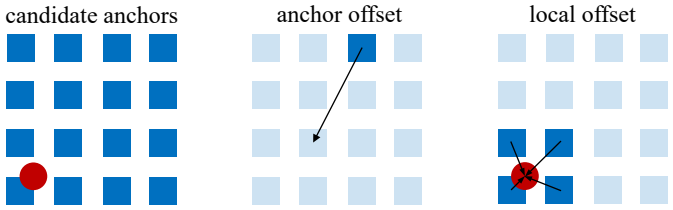

Figure 3: Illustration of the Local Refinement, the anchor offset gives a coarse guidance to the target, then we can refine it with a separately estimated local offset.

used to provide a coarse guidance to the target position, we denote it as $(\delta x, \delta y)$. The second is a local keypoint offset map that is estimated around the target keypoint, we can get the local offset $(\delta x_l, \delta y_l)$ by bilinear sampling on the coarse estimation $(i + \delta x, j + \delta y)$, thus the final estimation can be denoted as,

$$\mathcal{P}_r = \mathcal{A} + (\delta x, \delta y) + (\delta x_l, \delta y_l) \tag{12}$$

where $\mathcal{P}_r$ denotes the refined target estimation. We use it in equation 8 to get the regression score. The local offset branch is supervised by a separate loss. With local refinement, we can relieve the contradiction in naive anchor scoring and selecting, producing both high confident and accurate localization results. We will evaluate the importance of local refinement in Sec. 4.1.

4 Experiments on Single Keypoint Localization

We first test our method on single keypoint localization task, *i.e.*, single person pose estimation and facial landmark detection. There only exists one target for each type of keypoint in the input image, thus we only need to select the candidate with maximum confidence as final localization result. The datasets, evaluation metrics, training and test details of all tasks are provided in the appendix.

4.1 Ablation Study

We first evaluate each component of the proposed method on COCO dataset. We adopt the standard model SimplePose (Xiao et al., 2018) for experiments. For comparison, we first report the performance of several baselines, including model with regression-anchor and heatmap-anchor, and a baseline that adopts all the same setting of AdaAnchor except that the anchor is pre-defined and selected around the target, we denote it as FixAnchor. Their performance are shown in Table 1, we can observe that our method achieves smaller error on both visible and invisible keypoints. It reduces the error of invisible keypoints from 17.1 to 16.2 and improves the confidence from 0.568 to 0.805, which verifies that AdaAnchor can produce higher confident and accurate localization results.

We also investigate different ways to fuse two scores in Anchor Scoring. From Table 1 we can find that multiplication performs better than addition in all metrics, especially in invisible keypoints. This may be that 'mul' only output high score when both confidence and regression accuracy are large, while 'add' may only need one part, which is not as expected.

Finally we evaluate the two strategies used in AdaAnchor. We first test the effectiveness of multiple anchors. AdaAnchor without multiple anchors performs worse than heatmap-anchor on both invisible and visible keypoints. This indicates that providing more positive supervision is important to train localization model. Local Refinement is also an important component to boost the performance of localization error, reducing the error from 6.47 to 6.02 for visible keypoints and 16.8 to 16.2 for invisible keypoints. But it has less effect on confidence score, this demonstrates that local precise offset can relieve the long distance regression issue, which is important for accurate localization.

4.2 Single Person Pose Estimation

We evaluate the proposed AdaAnchor on single person pose estimation task. These experiments are conducted on large-scale in-the wild benchmark: COCO (Lin et al., 2014) and MPII (Toshev & Szegedy, 2014). We compare our method with widely used heatmap (Xiao et al., 2018) and regression (Toshev & Szegedy, 2014) under the same setting, *i.e.*, input size, output stride and backbone.

Table 1: Ablation study on COCO Keypoint. ResNet-50 is adopted as backbone.

Method	Vis.Error	Vis.Score	Inv.Error	Inv.Score	mAP
Baselines					
Regression-anchor	7.19	N/A	17.6	N/A	54.1
Heatmap-anchor	6.19	0.800	17.1	0.568	71.1
FixAnchor	6.17	0.709	17.0	0.511	71.1
Fusion strategy					
Scoring (add)	6.13	0.886	16.9	0.683	71.2
Scoring (mul)	6.02	0.925	16.2	0.805	71.3
Componet analysis					
Ours w/o multiple anchors	6.34	0.866	17.0	0.705	70.2
Ours w/o local refinement	6.47	0.902	16.8	0.783	69.0
Ours	**6.02**	**0.925**	**16.2**	**0.805**	**71.3**

Table 2: Comparison with other methods on COCO in single keypoint localization.

Method	Backbone	Vis.Error	Vis.Score	Inv.Error	Inv.Score	mAP
Regression-anchor	ResNet50	7.19	N/A	17.6	N/A	54.1
Heatmap-anchor	ResNet50	6.19	0.800	17.1	0.568	71.1
Ours	ResNet50	**6.02**	**0.925**	**16.2**	**0.805**	**71.3**
Regression-anchor	ResNet101	6.88	N/A	17.0	N/A	56.4
Heatmap-anchor	ResNet101	5.99	0.811	16.4	0.591	72.2
Ours	ResNet101	**5.94**	**0.918**	**16.2**	**0.805**	**72.4**
Regression-anchor	ResNet152	6.76	N/A	17.0	N/A	57.7
Heatmap-anchor	ResNet152	5.84	0.810	16.2	0.586	72.7
Ours	ResNet152	**5.76**	**0.929**	**15.5**	**0.820**	**73.0**

We first compare our method with heatmap and regression, previous two widely adopted methods in localization task. Because COCO provides the visibility of each keypoint, we further evaluate the prediction error on visible keypoints and invisible keypoints separately. Moreover, to reflect the importance of confidence score in localization, we report the mean score of the detected keypoints. All results are shown in Table 2. We can observe that our method outperform previous methods on all backbone and all metrics. Especially, our method reduces more error on invisible keypoint comparing with visible keypoint, *i.e.*, 0.17 v.s. 0.9. Moreover, our method output much higher confidence on invisible keypoint, 0.905 v.s. 0.568, therefore our method can avoid low confident localization results, which is more important when multiple targets exist. Fig. 4.1 shows some selected anchors by our method, we can observe that AdaAnchor can select different anchors to locate target. We also provide pose estimation results on MPII, these results are shown in Table 3, we can observe that our method achieves higher localization accuracy on PCKh@0.1 than heatmap.

Table 3: Comparison with other methods on MPII. ResNet-50 is adopted as backbone.

Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	PCKh0.5	PCKh0.1
Regression-anchor	95.1	93.9	81.7	72.8	84.8	67.0	58.0	80.4	20.5
Heatmap-anchor	96.6	94.6	88.2	82.6	87.6	83.0	78.2	**87.8**	33.6
Ours	96.6	94.8	88.0	81.2	87.6	82.3	77.4	87.4	**34.4**

4.3 Facial Landmark Detection

We also provide more experiments on facial landmark detection dataset WFLW (Wu et al., 2018). Normalized mean error (NME) is adopted as evaluation metrics. Quantitative results are reported in Table 4. Consistent with other experiments, our method provides performance improvement on facial landmark detection.

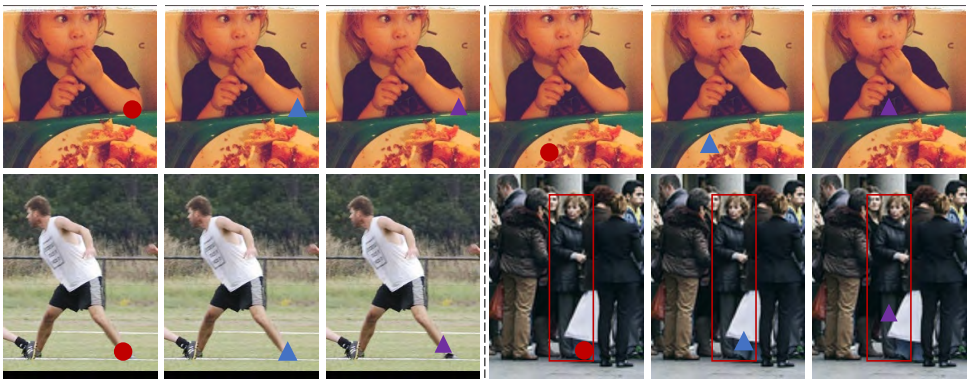

Figure 4: Visualization of selected anchors from COCO (Lin et al., 2014) `val` set. From left to right: target keypoint, heatmap-anchor and AdaAnchor. It can be observed that AdaAnchor can adaptively select anchor that is close to target (left) or in related visible region (right).

Table 4: Comparison with other methods on WFLW. HRNet-W18 is adopted as backbone.

Method	Test	LP	Expr.	Illu.	Mu.	Occu.	Blur
Regression-anchor	8.97	15.39	9.97	8.59	8.52	9.73	9.19
Heatmap-anchor	4.60	7.94	4.85	4.55	4.29	5.44	5.42
Ours	**4.44**	**7.44**	**4.78**	**4.38**	**4.25**	**5.22**	**4.99**

5 Experiments on Multiple Keypoint Localization

We then evaluate our method on multiple keypoint localization task, which is rarely exploited by previous methods. Only heatmap-based methods can be applied to this task, regression is not applicable due to the simple pre-defined anchor. Multiple keypoints in localization is common because it eliminates the assumption of single keypoint localization that only one instance exists in the input image. For instance, both bottom-up and single-stage multi-person pose estimation (Newell et al., 2017; Geng et al., 2021) require multiple keypoint localization. Confidence plays an equally important role as accuracy in multiple keypoint localization because we rely on it to select arbitrary number of results. Following previous work, we filter out predictions with confidence lower than γ.

5.1 Bottom-up multi-person pose estimation

Bottom-up multi-person pose estimation methods can be divided into two stages, the first is to detect multiple keypoints for all people in the image, then a grouping operation is applied to compose these keypoints into poses. Associative Embedding (AE) (Newell et al., 2017) is a typical bottom-up method that assign each keypoint with a tag, grouping can be done by clustering on the tags. Due to its simplicity and can be plugged into any keypoint localization method, we apply AdaAnchor into AE. To remove the influence of grouping operation on final localization performance, we use a pretrained Associative Embedding model to tag each predicted keypoints. Therefore our model only need to estimate keypoint locations. We set $\gamma = 0.1$ for all experiments.

We evaluate our method and heatmap-anchor on COCO and CrowdPose, the results are shown in Table 5 and Table 6. We can observe that our method achieves better performance on two dataset, *e.g.*, it outperforms heatmap by 4.8% AP on COCO[1]. Moreover, we also calculate the recall of detected keypoints, and our method has higher recall, which means AdaAnchor produces less missed detection results. Detailed computation process of recall can be found in the appendix.

[1]The mAP is lower than original AE model, *e.g.*, HigherHRNet in its paper, because we remove the post-processing refinement that fills the missed keypoints in the first stage for fair comparison. HigherHRNet without refinement achieves similar performance to heatmap, 47.8 and 45.7, but with a higher resolution feature map.

Table 5: Comparison with other methods on COCO in multiple keypoint localization.

Method	Recall	AP	AP.5	AP.75	AP^M	AP^L	AR
Heatmap-anchor	0.943	45.7	69.8	48.3	39.0	55.0	52.0
Ours	**0.986**	**50.5**	**73.1**	**53.8**	**42.0**	**62.8**	**55.4**

Table 6: Comparison with other methods on CrowdPose in multiple keypoint localization.

Method	Recall	AP	AP.5	AP.75	AP^E	AP^M	AP^H
Heatmap-anchor	0.803	41.8	65.2	41.4	55.3	41.4	33.7
Ours	**0.889**	**47.1**	**70.0**	**47.8**	**60.6**	**46.6**	**38.2**

5.2 Single-stage regression multi-person pose estimation

Single-stage regression methods, *e.g.*, CenterNet (Zhou et al., 2019) and DEKR (Geng et al., 2021) densely regress keypoints from person center points for multi-person pose estimation. Therefore multiple person center points localization is a core step in these methods, and they usually adopt heatmap to achieve this goal. Our method can also be applied to single-stage regression framework by replacing the heatmap with AdaAnchor to locate multiple center points.

We conduct experiments based on DEKR due to its superior performance. Experimental results are reported on CrowdPose (Li et al., 2019) and shown in Fig. 5.2 (b). Our method outperforms heatmap-anchor by 0.8% AP when $\gamma = 0.01$, the default setting of previous method, and shows constantly superior performance on larger γ, which further demonstrates that our method is more robust and can output high confidence localization results to avoid missed detection in crowded scenes.

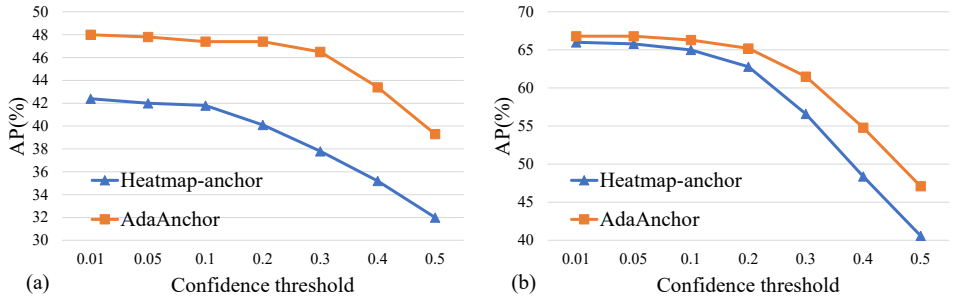

Figure 5: Confidence analysis on multiple keypoint localization tasks. (a) Bottom-up pose estimation with AE (Newell et al., 2017) on CrowdPose (Li et al., 2019). (b) Single-stage regression DEKR (Geng et al., 2021) on CrowdPose (Li et al., 2019). It can be observed that AdaAnchor outperforms heatmap-anchor on various confidence threshold, indicating that it produces higher confident and accurate localization results.

6 Conclusion

In this paper, we propose the Adaptive Anchor (AdaAnchor) to tackle the robust keypoint localization in various scenarios. Different from previous fixed anchor strategy, the key of the proposed method is to adaptively select anchors by maximizing both the confidence and accuracy of the final results. Benefited by this property, AdaAnchor achieves superior performance on both single keypoint and multiple keypoint localization tasks, avoiding low confident and missed detection due to occlusion and truncation.

REFERENCES

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

Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4733–4742, 2016.

Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *CVPR*, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 248–255. Ieee, 2009.

Zigang Geng, Ke Sun, Bin Xiao, Zhaoxiang Zhang, and Jingdong Wang. Bottom-up human pose estimation via disentangled keypoint regression. In *CVPR*, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *CVPR*, 2019.

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*, 2019.

Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11025–11034, 2021a.

Jiefeng Li, Tong Chen, Ruiqi Shi, Yujing Lou, Yong-Lu Li, and Cewu Lu. Localization with sampling-argmax. *Advances in Neural Information Processing Systems*, 34:27236–27248, 2021b.

Yanjie Li, Shoukui Zhang, Zhicheng Wang, Sen Yang, Wankou Yang, Shu-Tao Xia, and Erjin Zhou. Tokenpose: Learning keypoint tokens for human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11313–11322, 2021c.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Weian Mao, Zhi Tian, Xinlong Wang, and Chunhua Shen. Fcpose: Fully convolutional multi-person pose estimation with dynamic instance-aware convolutions. In *CVPR*, 2021.

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

Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *NeurIPS*, 2017.

George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *CVPR*, 2017.

George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *ECCV*, 2018.

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. In *NeurIPS*, 2019.

Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 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.

Dahu Shi, Xing Wei, Liangqi Li, Ye Ren, and Wenming Tan. End-to-end multi-person pose estimation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11069–11078, 2022.

Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, 2018.

Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *Advances in neural information processing systems*, 27, 2014.

Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1653–1660, 2014.

Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2129–2138, 2018.

Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018.

Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7093–7102, 2020.

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

A Appendix

In appendix we provide the detailed information of the datasets (A.1), evaluation metrics (A.2) and implementation details (A.3) of experiments. We also provide the analysis on multiple anchors k in equation 11 (A.4) and some qualitative results (A.5).

A.1 Datasets

COCO Keypoint (Lin et al., 2014) contains 64K images of 270K persons labeled with 17 keypoints. We use the `train` set containing 57K images, 150K persons for training. The `val` set containing 5K images, 6.3K persons and `test-dev` set containing 20K images are used for evaluation. We use this dataset for both single person and multi-person pose estimation tasks.

MPII (Toshev & Szegedy, 2014) consists of images taken from a wide-range of real-world activities with full-body pose annotations. There are around 25K images with 40K subjects, where there are 12K subjects for testing and the remaining subjects for the training set. We use this dataset only for single person pose estimation.

WFLW (Wu et al., 2018) is a dataset for facial landmark detection. There are 7,500 training and 2,500 testing images with 98 manual annotated landmarks. We report the results on the test set and several subsets: large pose (326 images), expression (314 images), illumination (698 images), make-up (206 images), occlusion (736 images) and blur (773 images). We use this dataset for facial landmark detection.

CrowdPose (Li et al., 2019) contains 20K images and 80K persons labeled with 14 keypoints. Following (Cheng et al., 2020; Geng et al., 2021), we use the `trainval` set (12K images, 43.4K

persons) and for evaluation we use the `test` set (8K images, 29K persons). We use this dataset for multiple keypoint localization task.

A.2 Evaluation Metrics

For each dataset, we follow previous works to report the official metrics, *i.e.*, mAP in COCO Keypoint, PCKh@0.5/0.1 in MPII and NME in WFLW.

Besides this, to give a more detail comparison of the proposed method in challenging scenarios, we further report the mean squared localization error between the prediction and groundtruth and corresponding confidence score on COCO Keypoint because it has visible and invisible annotation on each keypoint, which is denoted by *Vis.Error* and *Inv.Error* in the manuscript.

We also calculate the *Recall* in multiple keypoint localization evaluation. We follow the official implementation of mAP in `cocoapi` [2] to calculate recall. Given a set of detected keypoints $\{p_k\}_{k=1}^{m}$ and corresponding groundtruth $\{g_k\}_{k=1}^{n}$, where m and n denotes the number of keypoints in prediction and groundtruth. We first sort the predictions according to their confidence. Then we loop each prediction, find the groundtruth in the unmatched pool that is closest to it, then we mark this prediction and groundtruth as matched. After this matching process, suppose there are q matched gts, then the recall can be calculated by q/n.

A.3 Implementation Details

Here we provide detailed implementation for experiments on each task and dataset. All experiments are implemented on PyTorch (Paszke et al., 2019).

Single Keypoint Localization on COCO. We adopt SimplePose (Xiao et al., 2018) for ResNet-based experiments. We adopt various model as backbone, including ResNet-$\{50, 101, 152\}$. All model are pretrained on ImageNet. The input image is resized to 256×192. The learning rate is set to 1×10^{-3} at first and reduced by a factor of 10 at 90th epoch and 120 epoch. We use the Adam solver and train for 140 epochs, with batch size 320 in total. We adopt single scale test without flipping for all experiments.

Single Keypoint Localization on MPII. We adopt SimplePose (Xiao et al., 2018) for experiments. ResNet-50 is adopted as backbone, and is pretrained on ImageNet. The input image is resized to 256×256. The learning rate is set to 1×10^{-3} at first and reduced by a factor of 10 at 90th epoch and 120 epoch. We use the Adam solver and train for 140 epochs, with batch size 320 in total. We adopt single scale test without flipping for all experiments.

Single Keypoint Localization on WFLW. For WFLW, we follow previous work (Sun et al., 2019) and adopt HRNet-W18 (Sun et al., 2019) as backbone. The input image is resized to 256×256. The learning rate is set to 1×10^{-4} at first and reduced by a factor of 10 at 30th epoch and 50th epoch. We use the Adam solver and train for 60 epochs, with batch size 16 on one GPU in total.

Bottom-up Multiple Keypoint Localization on COCO and CrowdPose. We adopt HRNet-W32 (Sun et al., 2019) pretrained on ImageNet (Deng et al., 2009) as backbone for all experiments and follow the most configuration of HigherHRNet (Cheng et al., 2020). The input image is resized to 512*512. We uses Adam (Kingma & Ba, 2015) to optimize the model, and set the learning rate to 0.001 for all layers. We train the model for 140 epochs on COCO and CrowdPose, with learning rate dividing by 10 at 90th, 120th epoch. The batch size is set to 40 for CrowdPose and COCO. We adopt data augmentation strategies including random rotation (-30,30), scale ([0.75,1.5]), translation ([-40,40]) and flipping (0.5). During inference, we resize the short side of each image to 512, and keeps the aspect ratio. We adopt single scale test without flipping for all experiments.

Single-stage Person Center Localization on CrowdPose. We adopt HRNet-W32 (Sun et al., 2019) pretrained on ImageNet (Deng et al., 2009) as backbone and follow the most configuration of DEKR (Geng et al., 2021). The input image is resized to 512*512. We uses Adam (Kingma & Ba, 2015) to optimize the model, and set the learning rate to 0.001 for all layers. We train the model for 140 epochs on CrowdPose, with learning rate dividing by 10 at 90th, 120th epoch. The batch size

[2] `https://github.com/cocodataset/cocoapi/blob/master/PythonAPI/pycocotools/cocoeval.py`

is set to 40. During inference, we resize the short side of each image to 512, and keeps the aspect ratio. We adopt single scale test without flipping for all experiments.

A.4 Analysis on the number of multiple anchors k

We analyze the different number of k in equation 11 on the final localization performance, the results are shown in Fig. A.4. We test AdaAnchor with k from 1 to 50 and report the performance on both single keypoint localization (SimplePose) and multiple keypoint localization (AE) on COCO. The result indicates that $k > 5$ can constantly boost the performance, but not necessary for $k > 20$. Therefore we set $k = 20$ for all experiments.

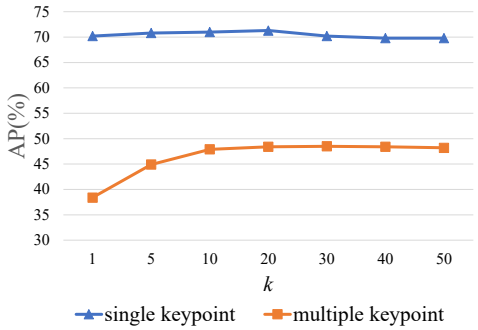

Figure 6: Analysis on different number of k in equation 11.

A.5 Qualitative Results

We provide qualitative results on human pose estimation and facial landmark detection estimated by our proposed AdaAnchor in Fig. A.5 and A.5.

Figure 7: Visualization results of human pose estimation on COCO (Lin et al., 2014).

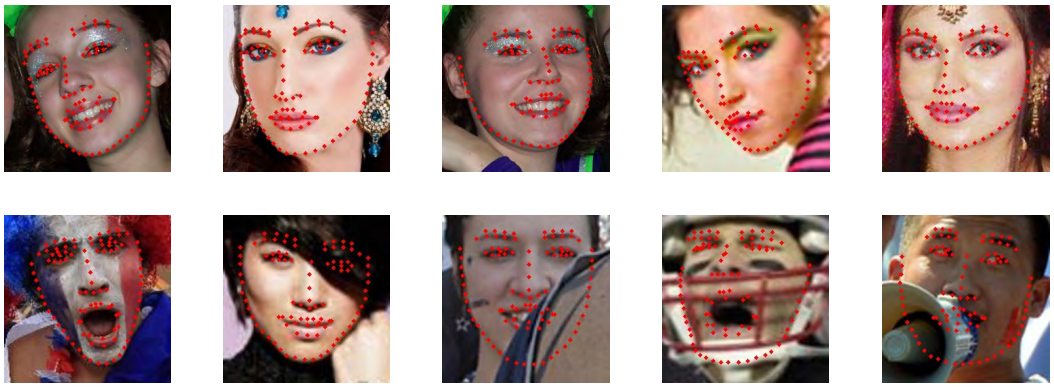

Figure 8: Visualization results of facial landmark detection on WFLW (Wu et al., 2018).