Removing the Bias of Integral Pose Regression

Kerui Gu[1] Linlin Yang[1,2] Angela Yao[1]
[1]National University of Singapore, Singapore
[2]University of Bonn, Germany
{keruigu, yangll, ayao}@comp.nus.edu.sg

Abstract

Heatmap-based detection methods are dominant for 2D human pose estimation even though regression is more intuitive. The introduction of the integral regression method, which, architecture-wise uses an implicit heatmap, brings the two approaches even closer together. This begs the question – does detection really outperform regression? In this paper, we investigate the difference in supervision between the heatmap-based detection and integral regression, as this is the key remaining difference between the two approaches. In the process, we discover an underlying bias behind integral pose regression that arises from taking the expectation after the softmax function. To counter the bias, we present a compensation method which we find to improve integral regression accuracy on all 2D pose estimation benchmarks. We further propose a simple combined detection and bias-compensated regression method that considerably outperforms state-of-the-art baselines with few added components.

1. Introduction

Pose estimation aims to determine the spatial position of articulated joints such as the human body or hand. At first glance, it seems that the problem should be a straight-forward regression. However, methods which directly regress joint coordinates [4, 29, 25] are less effective than those which locate the joints by estimating a likelihood heatmap [24, 8, 36, 28]. The rationale is that working with heatmaps allows the architecture to remain fully convolutional, thereby retaining spatial structures throughout the encoding and decoding process.

As labels, heatmap methods use a Gaussian centered at the ground truth joint coordinate. This formulation converts pose estimation into a detection problem; the network is tasked with predicting, at each pixel, the probability of that pixel being a joint pixel. The standard approach is to then use an argmax function to decode the output heatmap into a joint coordinate. The argmax function has two key drawbacks: it is not differentiable and it fixes the resolution of the estimated coordinate to that of the heatmap itself. Despite these disadvantages, however, detection-based methods[1] are highly effective and achieve state-of-the-art results for human pose estimation [24, 8, 36, 28].

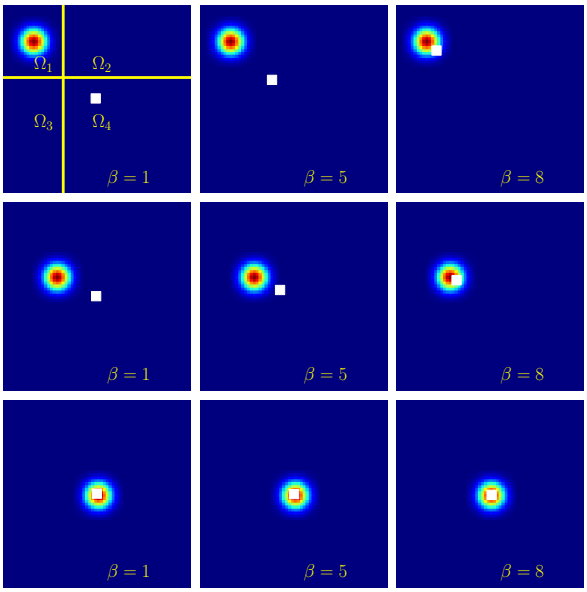

Figure 1. An illustration of the integral regression bias, with the implicit heatmap overlaid with the resulting predicted joint location as indicated by the white square. The bias, *i.e.* difference between the heatmap mode and predicted location, increases as the mode is further from the center and the value of β decreases. There is no bias only when the heatmap mode is centered. This paper proposes a method to compensate for the bias by partitioning the heatmap into individual regions (Ω_1 to Ω_4, see upper left heatmap) to ensure that the heatmap mode is centered.

An alternative form of decoding is to take a softmax together with an expectation [14, 29, 25, 21]. Both the soft-

[1]The naming convention used in the literature can be confusing in the context of this discussion as detection-based methods are often referred to as indirect regression or heatmap regression [2, 14, 40].

max and the expectation functions are differentiable. As such, one can train end-to-end directly from ground truth joint coordinates while retaining the benefits of fully convolutional architectures. This approach is referred to as latent heatmap regression [14] in the hand pose estimation literature and integral pose estimation [29] in body pose estimation literature.

Sun *et al*. [29]'s comparison between integral pose regression versus detection-based heatmaps concluded that integral regression is either as competitive or better for 2D pose estimation. However, the results may not be fully conclusive, as [29] used a unique backbone for integral regression when comparing against existing detection-based methods. Given that detection-based methods still dominate state-of-the-art, this begs a revisiting of the question. Which is better for 2D pose estimation - heatmap based detection or integral regression?

A critical difference when comparing the two methods is that integral regression method is slower to converge than heatmap-based detection. [29, 40] have put forth that the explicit heatmap label in detection provides a denser form of supervision than the joint coordinates of integral regression. We show, as a main contribution of this paper, that the slow convergence may also be the result of an induced bias in integral regression. Specifically, the combination of the softmax with an expectation shifts the heatmap's alignment with respect to the true coordinate position. This in turn limits the ability of the neural network to learn efficiently. By compensating for the bias, we improve both the learning and performance of the integral regression method.

In our performance comparisons, we are inspired by [27] to dig deeper into the factors of variation present in current benchmarks. To that end, we go beyond the current standard of reporting of a single (average) value of AP and PCK to separate different factors. Surprisingly, we find that in the "hard" cases of pose estimation, *i.e*., smaller size, fewer joints, or more occlusion, integral regression outperforms heatmap based detection. However, these effects are obscured because the hard cases constitute the dataset tail.

Finally, in an effort to retain the benefits of both detection-based heatmaps and integral regression, we propose a combined learning framework. Specifically, we incorporate our bias compensation into the end-to-end learning of integral regression, while leveraging the detection-based loss in initial training epochs to leverage the dense supervision. Summarizing our contributions,

- We show via derivation a previously unobserved bias in integral pose regression based on the combination of the softmax and expectation operations.
- We propose a simple compensation scheme to counter this bias which can up training and improve the performance of integral regression methods.
- We analyze the performance differences of detection- and regression-based human pose estimation methods for different factors of variation. Regression method performs better in "hard" cases of fewer joints, more occlusion and lower-resolution inputs.
- We propose a bias-compensated joint framework for detection and regression that retains the benefits of both fast convergence and high performance in hard cases. Our new framework achieves state-of-the-art results on MS COCO and MPII for 2D human pose estimation and RHD for 2D & 3D hand pose estimation.

2. Related Work

Regression-based methods. Classical methods [34, 4] used CNNs to extract features and directly predict joint locations with fully connected layers. Shortly after, integral pose regression [29] was proposed; it can leverage the use of a fully convolutional architecture while training end-to-end from joint coordinates. The "integral" takes on the form of a soft-argmax with an expectation and has also been used in previous works [16, 39, 30]. While there are a large number of detection-based methods, only two recent works [21, 25] followed an integral regression approach. [21] combined contextual information with a regression loss while [25] proposed a variance or distribution penalty on top.

Detection-based methods. Since the heatmap representation was introduced in [33], heatmap-based detection methods [11, 35, 18, 6, 9, 38, 37, 20, 7] has dominated human pose estimation. A well-known example is the Hourglass Network [24], which stacks together encoder-decoder modules with skip connections to estimate and gradually refine the joint heatmaps. More recently, the Simple Baseline (SBL) [36] proposed a simple but effective baseline by adding a few deconvolutional layers. HRNet [28] improved the performance with a higher-resolution framework. Other work [23] adds additional modules on top of HRNet to improve the performance by considering the neuroevolution.

In contrast to all the above works, our motivation is not to develop a new architecture for human pose estimation. Instead, we investigate the underlying drawbacks of regression-based human pose estimation and show that a possible cause could be a bias in combining the softmax with the expectation. Our work is analogous to DARK [40] and UDP [13], which are designed for detection-based pose estimation and address the quantisation error and biased data processing respectively.

3. Pose Estimation Preliminaries

We consider top-down approaches in which already-cropped image $\mathbf{I}$ of a person is provided. The aim of 2D body pose estimation is to predict the 2D coordinates $\mathbf{J} \in \mathbb{R}^{K\times 2}$ of K body joints. The standard in 2D pose

estimation is to use an encoder-decoder framework. The encoder, via a series of convolutional layers, reduces the image to a low-resolution feature map; the decoder then progressive up-samples the feature maps into a final output $\mathbf{H} \in \mathbb{R}^{h \times w \times K}$, where h and w is a scale factor of the shape of the input image I and each of the K channels represents the output for a specific joint k. For clarity, we drop the subscript k in our exposition below and use simply $\mathbf{H}$ and $\mathbf{J}$ to represent the heatmap and coordinates of joint k.

3.1. Detection-Based Method

In detection-based methods, $\mathbf{H}$ is treated as an explicit heatmap. During inference, the coordinates of joint k, $\mathbf{J}^d$, is estimated by taking an argmax operation,

$$\mathbf{J}^d = \arg\max_{\mathbf{p}} \mathbf{H}(\mathbf{p}), \tag{1}$$

where $\mathbf{p}$ denotes the pixel coordinate in the heatmap. This formulation can be interpreted as taking a maximum likelihood on the heatmap if $\mathbf{H}$ was proportional to the probability density of the joint location. In practice, however, the final coordinate estimate is actually shifted to sit between the highest and second-highest response on the heatmap to account for the heatmap quantization [24].

During training, a ground truth heatmap $\mathbf{H}^{gt}$ is generated for joint k by placing a small Gaussian centered at the ground truth coordinate $\mathbf{J}^{gt}$. The loss applied for k^{th} joint is a pixel-wise MSE between $\mathbf{H}^{gt}$ and estimated heatmap $\mathbf{H}$:

$$L_{\text{de}} = \frac{1}{|\Omega|} \sum_{\mathbf{p} \in \Omega} \|\mathbf{H}(\mathbf{p}) - \mathbf{H}^{gt}(\mathbf{p})\|^2, \tag{2}$$

where Ω denotes the set of all pixel locations in the heatmap and $|\Omega|$ denotes the number of pixels in the heatmap.

3.2. Integral Regression Method

In the integral regression method, $\mathbf{H}$ is not an explicit heatmap and hence the "latent" demarcation by [14]; it is decoded into coordinates by applying a softmax normalization followed by an expectation operation. More specifically, the estimated heatmap $\mathbf{H}$ for joint k is normalized to $\tilde{\mathbf{H}}$ via a softmax function:

$$\tilde{\mathbf{H}}(\mathbf{p}) = \frac{\exp(\beta \cdot \mathbf{H}(\mathbf{p}))}{\sum_{\mathbf{p}' \in \Omega} \exp(\beta \cdot \mathbf{H}(\mathbf{p}'))}, \quad \beta > 0 \tag{3}$$

where β is a smoothing parameter. The softmax ensures that the elements of $\tilde{\mathbf{H}}$ will sum up to 1 so that $\tilde{\mathbf{H}}$ can be applied directly as a probability density when taking the expected value to estimate the coordinate $\mathbf{J}^r$:

$$\mathbf{J}^r = \mathbb{E}_{\tilde{\mathbf{H}}}[\mathbf{p}] = \sum_{\mathbf{p} \in \Omega} \tilde{\mathbf{H}}(\mathbf{p}) \cdot \mathbf{p}. \tag{4}$$

The key advantage of taking the expected value instead of the argmax is that it is a differentiable operation. For training, an L1 distance between the joint coordinates for each joint can be applied directly as a loss:

$$L_{\text{re}} = \|\mathbf{J}^r - \mathbf{J}^{gt}\|. \tag{5}$$

The L1 is preferred over L2 as a loss because of its better performance; this was first studied in [29].

4. Bias-Compensated Integral Pose Regression

4.1. Derivation of Bias

To take the expectation, we need a normalized probability density function and the softmax function serves that purpose to normalize $\mathbf{H}$. However, the softmax is also dense in that it assigns nonzero values to *all* the pixels in $\tilde{\mathbf{H}}_k$, even for the zero elements of $\mathbf{H}$[2]. The non-zero assignments to the (close-to) zero-valued pixels of $\mathbf{H}$ in turn contributes to the expected value and biases the estimated coordinate $\mathbf{J}^r$ towards the center of the heatmap. The further away the joint coordinate is from the center, the greater the bias (see Fig. 1).

The effects of such a bias can be alleviated somewhat by choosing an appropriate value for β in the softmax. The smaller β is, the more the function distributes the probability density and the greater the impact of the zero-pixels in $\mathbf{H}$. The larger β is, the more the function concentrates the density around the largest values of $\mathbf{H}$. In the limit when β goes to infinity, the softmax converges to the argmax function [5]. In turn, it also becomes non-differentiable. In fact, as β gets progressively larger, the gradients of pixels which are further away from the center become smaller and gradually approach zero. It is therefore necessary to trade off between the extent of the bias versus having sufficient gradients for learning.

Note that the decoder can compensate for this bias by learning to estimate an $\mathbf{H}$ with densities shifted even further away from the center. We posit that this adds an extra learning challenge for the neural network, since it must now account for the displacement of the joints from the heatmap center. When looking at the distribution of joint locations, *e.g.* of the MS COCO training dataset (see Fig. 2), we can see that this adds a significant source of additional variation of bias values. This is likely further exacerbated by the data augmentation, which is standard practice for training.

4.2. Bias Compensation

As shown in Fig. 1, the only case in which there is no bias is when the probability density is centered on the heatmap. A naïve way to to compensate for the bias to directly shift the coordinate system and center the heatmap at the ground

[2] Consider the numerator of Eq. (3), $\exp(\beta \cdot 0) = 1$

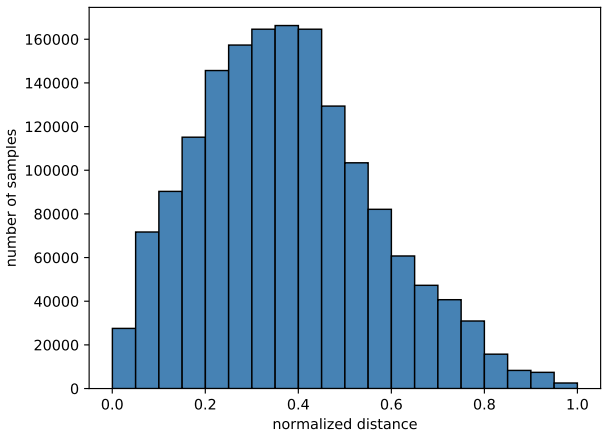

Figure 2. Distribution of joint locations on training set of COCO. The distance is normalized by the maximum joint distance to the center. For $\beta = 10$ (see Eq. 3), normalized joint distances of $\{0.2, 0, 5, 0.8\}$ from the center result in biases of $\{0.2, 0.7, 1.2\}$ px respectively of a 64×48 heatmap. Using a smaller β makes this bias even more extreme.

truth coordinate $\mathbf{J}^{gt}$; then, there would be no bias when taking the expectation of $\tilde{\mathbf{H}}$. However, this requires knowing the location of $\mathbf{J}^{gt}$, which is feasible for training, but not suitable at inference time.

As such, we propose a bias compensation scheme that removes the contributions of the additional support encoded in $\mathbf{H}$, *i.e.* extra non-zero assignments. Suppose we wish to recover the true coordinate location (x_o, y_o); assume for now that (x_o, y_o) lies in the upper left quadrant of the heatmap. We can partition the image plane into four rectangular sections with splits at $2x_o$ and $2y_o$, with Ω_1 as the section that contains (x_o, y_o) and Ω_2, Ω_3 and Ω_4 denoting the other regions in clockwise fashion (see the top-left image in Fig. 1). Based on this partition, we can split the expectation defined in Eq. (4) as follows

$$\mathbf{J}^r = \sum_{\mathbf{p}\in\Omega_1} \tilde{\mathbf{H}}(\mathbf{p}) \cdot \mathbf{p} \;+\; \sum_{\mathbf{p}\in\Omega_2,\Omega_3,\Omega_4} \tilde{\mathbf{H}}(\mathbf{p}) \cdot \mathbf{p}. \tag{6}$$

We assume that the support for (x_o, y_o) is well localized, *i.e.* fully contained within Ω_1 in $\mathbf{H}$ and that sections Ω_2 to Ω_4 contain only zero or near-zero elements. As such, only the first term of Eq. (6) should contribute to the $\mathbf{J}^{ro}$. It follows then that the joint location can be estimated as a scaled version of the first term of Eq. (6)

$$\mathbf{J}^{ro} = \frac{1}{w_1} \sum_{\mathbf{p}\in\Omega_1} \tilde{\mathbf{H}}(\mathbf{p}) \cdot \mathbf{p}, \text{ where } w_1 = \sum_{\mathbf{p}\in\Omega_1} \tilde{\mathbf{H}}(\mathbf{p}), \tag{7}$$

where we define $\mathbf{J}^{ro}$ as the estimate of (x_o, y_o), *i.e.* a bias-compensated joint location. Note that the above formulation is implicit since Ω_1 depends on (x_o, y_o). With algebraic rearrangement, we can formulate $\mathbf{J}^{ro}$ as a function of $\mathbf{J}^r$:

$$\mathbf{J}^{ro} = \frac{C}{(C-wh)}\mathbf{J}^r - \begin{bmatrix} \frac{hw^2}{2(C-wh)} \\ \frac{h^2w}{2(C-wh)} \end{bmatrix}. \tag{8}$$

C above is the normalizing constant used in the softmax, *i.e.* the denominator of Eq. (3), and is a function of β:

$$C(\beta) = \sum_{\mathbf{p}\in\Omega} \exp(\beta \cdot \mathbf{H}(\mathbf{p})). \tag{9}$$

From Eq. (8) and Eq. (9), we see that the impact of the bias is negligible for a large C, since the scaling factor approaches one while the offset will approach zero. This is exactly the case when a large β is used, *i.e.* the softmax approaches the argmax function. However, when C is small, then the bias becomes more significant; so if β is not sufficiently large, the network must learn very large and concentrated values of $\mathbf{H}(\mathbf{p})$ to compensate to estimate the correct $\mathbf{J}^r$. We posit that it is exactly this interplay which makes it very challenging for the network to learn and hence the slow convergence rates of the integral regression method (see Sec. 5.4). We refer the reader to the Supplementary for the full derivation as well as other cases when x_o, y_o are in the other quadrants.

From Eq. (8), we can recover the bias-compensated joint location $\mathbf{J}^{ro}$. Note that this formulation does not require knowledge of the ground truth. During inference, we can directly compensate for the biased location $\mathbf{J}^r$ based on the expectation in Eq. (4). During training, we can do the same, and simply update the L1 loss of Eq. (5) with $\mathbf{J}^{ro}$, *i.e.*

$$\mathcal{L}_{\text{re}} = \|\mathbf{J}^{ro} - \mathbf{J}^{gt}\|. \tag{10}$$

In our approach, we have opted to retain the softmax and instead compensate for the expectation. Naive activation functions in place of the softmax have been explored for human pose estimation, but have been shown to be less effective [25]. A less naive option is the sparsemax [22], which has been proposed as a sparse alternative to the softmax. It projects the pre-activation value to a simplex so only a few non-zero values are preserved. However, given the large size of the flattened heatmap, only assigning a number of pixels with non-zero values makes it hard to train.

4.3. Joint Framework

We now propose a new method for pose estimation that incorporates our bias compensation. We wish to have the advantage of end-to-end training from integral pose regression, but still retain the fast training speeds of heatmap-based detection. It is believed that using the Gaussian heatmap as a label in detection methods can provide dense

supervision and spatial information which makes the training effective and efficient. However, as studied in [25], the pixel-wise L2 loss enforces the heatmap to be exactly same as the ground truth. This is a metric that we do not really care about because the loss cannot guarantee that the accuracy of joint predictions will always improve.

We therefore use the pixel-wise supervision in the detection-based methods (Eq. (2)) as an additional loss:

$$\mathcal{L} = \mathcal{L}_{\text{re}} + \lambda(t) \cdot \mathcal{L}_{\text{de}}, \tag{11}$$

and reduce λ over time. This has the effect of minimizing the impact $\mathcal{L}_{\text{de}}$ in subsequent epochs and allows the network to learn the arbitrary shape of the implicit heatmap. Out of simplicity, we use a simple step function for $\lambda(t)$, *i.e.* $\lambda(t) = 1$ for $t < T_o$ and $\lambda(t) = 0$ for $t >= T_o$. The key purpose of this joint framework is to speed up training while maintaining the freedom of the implicit heatmap to allow the network to learn any distribution (rather than a Gaussian) that can lead to the correct coordinate location.

5. Experiments

5.1. Datasets and Implementation Details

Datasets and evaluation metrics. We evaluate our methods on two human pose datasets, MS COCO [19], MPII [1] and one hand pose dataset, RHD [41].

The **COCO dataset** has 250k person instances with 17 annotated keypoints. We evaluate with the standard metric, Object Keypoint Similarity (OKS). OKS utilizes the area of person instance to normalize the absolute error between the predicted location and ground truth location. We use the main competition metric, the mean average precision(AP) over 10 OKS thresholds to evaluate the performance. We also report the value before normalization, the squared Euclidean distance between the prediction and ground truth, which we denote as End Point Error (EPE).

The **MPII dataset** contains 49k person instances with 16 annotated keypoints. We use the standard train/val split of [33] and evaluate performance with Percentage of Correct Keypoints (PCK) and EPE.

RHD is a synthesized hand dataset with 41k training and 2.7k testing images from 20 animated characters. For each RGB image, 21 hand keypoint annotations are provided. We follow [14] and evaluate with AUC and EPE.

Implementation details. We implement our experiments in Pytorch [26] and use Adam [15] for the model training. We use SBL [36] and HRNet [28] with different backbones, *e.g.*, SBL-ResNet50 and HRNet-W32, as our baselines and follow the same learning configurations for detection-based, regression-based, and our method. We rerun the experiments, *i.e.*, detection-based methods, to report the results for a fair comparison. We set $\beta = 10$ as default value of integral regression method. For our combined method, we use a T_o of 120 epochs for SBL and 190 epochs for HRNet.

5.2. Performance Comparison

5.2.1 Architecture design

To directly compare detection- and regression-based methods, other influencing factors affecting the final performance should be excluded. Generally, the framework of both methods are the same, in that an encoding step extracts features from input images while a decoding step converts the features back into an explicit or implicit heatmap representation. The difference lies only in the loss. The detection methods use an MSE Loss to supervise the explicit heatmap, while regression methods directly supervise the predicted joint location generated by taking the expectation of the implicit heatmap representation.

Former research [36, 28, 25] has shown that powerful encoders which are better in extracting features boost prediction accuracy. Meanwhile, it's also straightforward to understand that different decoders, *i.e.*, different numbers and parameters of upsampling layers, will also affect the results. In addition, representation resolution, including input size and heatmap size, is another key variable. Therefore, we adopt the same encoding-decoding architecture and representation size to create a fair comparison between detection- and regression-based methods.

When implementing the comparison, we apply the off-the-shelf models, *i.e.* SBL [36] or HRNet [28], as the shared encoding-decoding architecture to generate the heatmap $\mathbf{H}$ all K joints. For detection, we train the model by Eq. (2) and make final predictions by the argmax in Eq. (1). For regression, we apply the state-of-the-art regression method [29], *i.e.* integral regression to obtain the joint location by Eq. (4) and train the network by Eq. (5).

5.2.2 Influencing Factors

When the capacity of both methods, including extracting and representing features, is the same, the difference between the two only lies in the way of converting heatmap to coordinate and supervisory signals. Therefore, the results that the two methods present during the training and inference process can be compared to study the differences.

Specifically, in the training, we compare the two methods by evaluating the rising speed on the validation or test set. In the inference, we compare the generalization ability on different kinds of samples on the validation or test set. We divide the benchmarks according to the different layouts of people portrayed in the images. Similar with [27], three factors are taken into consideration: the amount of present joints or keypoints (11-17, 6-10, 1-5), the percentage of occlusion ($<10\%$, 10-50%, $>50\%$), and the largest dimension

Figure 3. We separate the COCO dataset based on input (bounding box) size, number of joints present in the scene, and percentage of occlusion (of the present joints). The splits from left to right roughly correspond to the difficulty of the pose estimation.

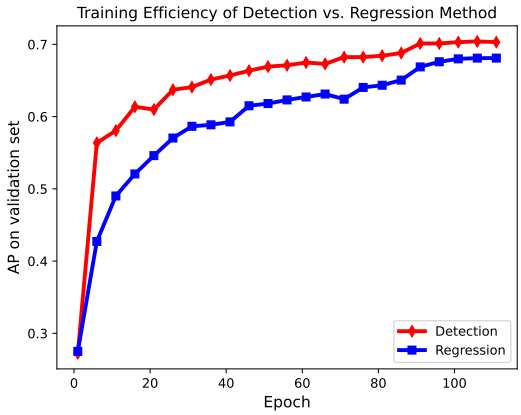

Figure 4. Although the detection and regression methods eventually converge to similar values, the detection-based method is more efficient in training, especially in the initial epochs.

of the bounding box input (> 128px, 96-128px, 64-96px, 32-64px). Examples of the divisions for these three factors are shown in Fig. 3.

5.2.3 Comparison Results

SBL-ResNet50 and HRNet-W32 are used as baselines to evaluate the training efficiency and generalization errors on COCO validation set.

Training comparison. Training efficiency can be evaluated by the ascending speed of AP as iteration increases. From Fig. 4, we can observe that, in the end, the detection and regression method reaches a similar accuracy. However, after 40 epochs, the detection method has already had a comparable result while the regression method needs 90 epochs to achieve the similar performance.

	# Joints			
Input Size	[11, 17]	[6, 10]	[0, 5]	all
[128^2,]	**8.89** / 9.16	13.96 / **13.82**	**33.0** / 33.8	**13.4** / 13.6
[96^2, 128^2]	**6.35** / 6.59	**10.9** / 11.0	21.1 / **17.2**	7.26 / **7.18**
[64^2, 96^2]	**4.41** / 4.57	9.12 / **9.05**	16.0 / **14.8**	7.26 / **7.18**
[32^2, 64^2]	**2.97** / 3.01	**4.32** / 4.51	9.23 / **8.1**	4.46 / **4.35**
%Occlusion	[0, 5]	[6, 10]	[11, 17]	all
> 50%	16.6 / **15.2**	16.0 / **14.8**	32.0 / **28.1**	19.0 / **17.4**
[10%, 50%]	**7.02** / 7.36	**6.88** / 7.00	23.7 / **22.8**	**8.36** / 8.53
< 10%	**4.91** / 5.22	**6.78** / 7.18	27.1 / **24.3**	**5.59** / 5.80
all	**6.91** / 6.96	**8.04** / 8.12	28.0 / **25.4**	**8.21** / 8.28

Table 1. Comparisons of EPE on the COCO validation set with a common SBL backbone, with separation according to number of present joints, input size, and percentage of occlusions. Presentation format is detection / regression. Regression outperforms detection with fewer joints present, smaller input sizes and more occlusion, though this phenomenon is obscured once all the factors are averaged due to the dataset distribution.

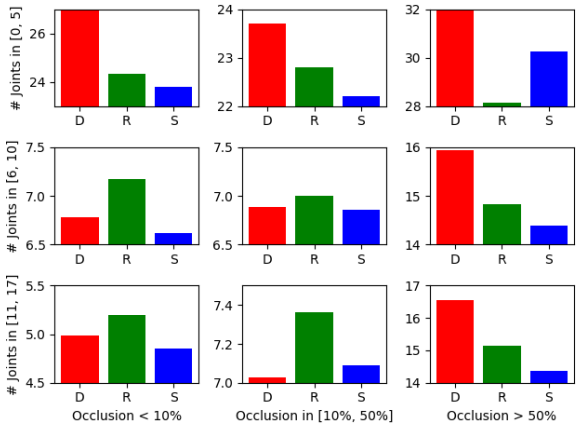

Figure 5. Comparisons of EPE of our method (S) with detection (D) and regression (R) on the divided sub benchmarks. Our method performs best *i.e.* has the lowest EPE in 7 of the 9 conditions.

We posit that one of the reasons is that in detection-based methods, dense supervision of full spatial map is applied; whereas in regression, the model only has supervision of only the expected coordinate location. Therefore, the latent feature map of regression methods can be arbitrary, as long as the final value corresponds to the correct joint location. Though this arbitrariness can be interpreted as beneficial, it does increase the difficulty in the training process. We demonstrate this by ablation study in Sec. 5.4.

Inference comparison. In Table 1, we report EPE on the divided benchmarks *w.r.t.* the amount of present keypoints, the size of input person instance, and the percentage of occlusion. If we only focus on the influence that different sizes have on the performance, the last column in the table shows that the detection performs better in large inputs and few occlusion cases, which can be considered easy, and regres-

Method	Type	Backbone	Input size	AP(%)↑	EPE(px)↓
Mask-RCNN [12]	D	ResNet-50-FPN	-	62.9	-
Hourglass [24]	D	8-stage Hourglass	256 × 192	66.9	-
CPN [8]	D	ResNet-50	256 × 192	71.6	-
IPR [29]	R	ResNet-101	256 × 256	67.2	9.98
+ Ours	R	ResNet-101	256 × 256	**69.1(+1.9)**	**9.42(-0.56)**
SBL [36]	D	ResNet-50	256 × 192	70.5	9.52
+ IPR	R	ResNet-50	256 × 192	68.2	9.63
+ Ours	R	ResNet-50	256 × 192	**71.2(+0.7)**	**8.93(-0.70)**
SBL [36]	D	ResNet-152	384 × 288	73.8	8.21
+ IPR	R	ResNet-152	384 × 288	71.3	8.28
+ Ours	R	ResNet-152	384 × 288	**74.4(+0.6)**	**7.82(-0.39)**
HRNet [28]	D	HRNet-W32	256 × 192	75.3	7.85
+ IPR	R	HRNet-W32	256 × 192	72.9	8.03
+ Ours	R	HRNet-W32	256 × 192	**75.8(+0.5)**	**7.47(-0.38)**

Table 2. Evaluation of our method competing with state-of-the-art methods on COCO validation set. 'D' and 'R' stands for detection- and regression-based methods respectively. Our proposed method outperforms both detection- and regression-based baselines.

Method	Type	Backbone	Input size	# Params	GFLOPS	AP (%)↑	AR (%)↑
Mask-RCNN [12]	D	ResNet-50-FPN	-	-	-	63.1	66.5
CPN [8]	D	ResNet-Inception	384 × 288	-	-	72.1	78.5
RMPE [10]	D	PyraNet	320 × 256	28.1M	26.7	72.3	-
SBL [36]	D	ResNet-152	384 × 288	68.6M	35.6	73.7	79.0
HRNet [28]	D	HRNet-W48	384 × 288	63.6M	32.9	75.5	80.5
MSPN [17]	D	4-stg MSPN	384 × 288	-	-	76.1	81.6
DARK [40]	D	HRNet-W48	384 × 288	63.6M	32.9	76.2	81.1
UDP [13]	D	HRNet-W48	384 × 288	63.8M	33.0	76.5	81.6
DirectPose [31]	R	ResNet-101	-	-	-	63.3	-
IPR [29]	R	ResNet-101	256 × 256	45.0M	11.0	67.8	-
Ours	R	HRNet-W48	384 × 288	63.6M	32.9	76.1	81.0

Table 3. Evaluation of our method competing with state-of-the-art methods on COCO test-dev set. 'D' and 'R' stands for detection- and regression-based methods respectively. Our proposed method is competitive against state-of-the-art detection-based methods and surpasses the performance of regression-based methods by a large margin.

sion performs better in tiny poses and many occlusion cases, which are harder; In addition, when the number of present keypoints changes, the performance of the two methods differs. Specifically, for easy cases, detection method performs better; for medium cases, regression method becomes competing; for hard cases, regression method even exceeds. We report more experiments in the supplementary material.

When adding our components, we extract both benefits of the detection- and regression-based methods. Experimentally, from Fig. 5, we can see that our method outperforms the regression method in hard cases and the detection method in easy cases.

5.3. Comparison with State-of-the-Art

Evaluation on MS COCO. We compared our method with top-performers of 2D human pose estimation models in Table 2 and Table 3 on COCO val and test-dev set. We can see that our method is competitive against state-of-the-art detection-based method and surpass the performance of regression-based methods by a large margin.

Evaluation on MPII. In Table 4, we also compare our method with state-of-the-art models on the MPII validation set with former regression-based methods, including [32], DSNT [25] IPR [29], and detection-based methods, including SimpleBaseLine [36] and HRNet [28].

Evaluation on RHD. We compare our method with a regression-based method, 2.5D regression [14] and two detection-based methods [41, 3] in Table 5. We can see that our method surpasses the baseline significantly with bias compensated and augmentation loss added.

We report the results of different methods in Table 6. The results show that the detection loss not only speeds up the training, but also enhances the performance. The EPE on hard samples demonstrates the superiority of our method over detection based one.

5.4. Ablation Study

Our method consists of two novel components: bias compensation and a regularization term. In this subsection, we do ablation studies to demonstrate the influence of each

Method	Type	PCKh@0.5(%)↑	EPE(px)↓
Tompson et al. [32]	R	80.2	-
DSNT [25]	R	85.7	-
IPR [29]	R	86.5	-
+ Ours	R	**87.2(+0.7)**	-
SBL-ResNet50 [36]	D	87.6	20.9
+ IPR	R	86.2	21.5
+ Ours	R	**87.9(+0.3)**	**20.3(-0.6)**
SBL-ResNet152 [36]	D	89.6	18.3
+ IPR	R	87.9	19.5
+ Ours	R	**89.9(+0.3)**	**17.8(-0.5)**
HRNet-W32 [28]	D	90.4	16.6
+ IPR	R	88.7	18.2
+ Ours	R	**90.6(+0.2)**	**16.2(-0.4)**

Table 4. Comparison on MPII validation set. Our method gains significant improvement on the baselines.

Method	Type	AUC(%)↑	EPE(px/mm)↓
Z&B [41]	D	72.0/67.5	9.14/30.4
Cai [3]	D	-/88.7	-/-
2.5D regression [14]	R	84.4 / 93.0	4.76 / 14.3
+ Ours	R	**85.8 / 93.6**	**4.34 / 13.5**

Table 5. Comparison on the RHD test set of 2D/3D AUC and EPE for hand pose estimation. Our proposed method outperforms the baseline and two detection-based methods.

Method	β	AP	EPE	EPE^{H}
SBL [36]	-	70.5	9.52	32.1
+IPR	10	68.2	9.63	28.7
$+\mathcal{L}_{\text{de}}$, $\lambda=1$ all epochs	10	70.2	9.38	28.0
$+\mathcal{L}_{\text{de}}$	10	70.4	9.31	27.8
+de-bias	10	71.0	9.15	27.1
$+\mathcal{L}_{\text{de}}$+de-bias	1	70.7	9.02	26.1
$+\mathcal{L}_{\text{de}}$+de-bias	20	70.3	9.32	30.7
$+\mathcal{L}_{\text{de}}$+de-bias	10	**71.2**	**8.93**	**25.6**

Table 6. Evaluation of each component of our method on COCO validation set. EPE^{H} denotes EPE on 'hard' samples, *i.e.* those with few number of joints present ([0,5]) and heavy occlusions(>50%). For $\mathcal{L}_{\text{de}}$, $\lambda(t) = 1$ for $T_o = 120$ unless otherwise indicated. Our proposed components improve the performance with respect to the baseline, especially on these hard samples. A combination with $\beta=10$ is optimal.

component. We use SBL-ResNet 50 as our backbone to perform the ablation experiments and the input size is 256 × 192. The results are evaluated on the COCO validation set.

Effectiveness of components. We evaluate the components by comparing the convergence speed of different methods and the inference results. Baselines are the detection-based and regression-based (+IPR) methods and we compare them with the models combined with detection loss or bias compensation. In addition, we also include the experiment when the regularization term is stronger and detection loss is applied for all epochs. We illustrate the curves of methods in Fig. 6, which stand for the AP on the validation set after training specific epochs. We can see that each component will speed up the training. Meanwhile, in Table 6, each component contributes to the higher performance.

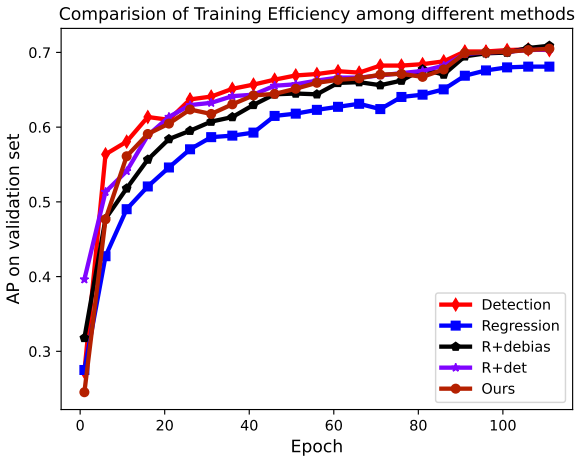

Figure 6. The influence of each component on the convergence speed on the COCO validation set. Our proposed components will accelerate the training speed of regression method, approaching to that of the detection method.

Selection of hyperparamter β. As illustrated above, a small β leads to large bias and a large β makes back propagation difficult. We evaluate the different selection of the value of β in Table 6 and choose $\beta = 10$ as our optimal value.

6. Conclusion

Our main contribution is to reveal the bias of integral pose regression method. We also, for the first time, systematically make a fair comparison between detection and regression-based methods. Inspired by the two, we propose to integrate the two methods to extract both benefits and compensate the bias in the integral regression. Experimental results on COCO, MPII, and RHD show that our method, which can be served as a plug-in component, improves the baseline by a large margin and proves to be effective in both 2D human pose estimation and hand pose estimation.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, pages 3686–3693, 2014. 5

[2] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *ECCV*, pages 717–732. Springer, 2016. 1

[3] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV*, pages 666–682, 2018. 7, 8

[4] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *CVPR*, pages 4733–4742, 2016. 1, 2

[5] Olivier Chapelle and Mingrui Wu. Gradient descent optimization of smoothed information retrieval metrics. *Information retrieval*, 13(3):216–235, 2010. 3

[6] Xianjie Chen and Alan Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. *arXiv preprint arXiv:1407.3399*, 2014. 2

[7] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *ICCV*, pages 1212–1221, 2017. 2

[8] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, pages 7103–7112, 2018. 1, 7

[9] Xiao Chu, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Crf-cnn: Modeling structured information in human pose estimation. *arXiv preprint arXiv:1611.00468*, 2016. 2

[10] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *ICCV*, pages 2334–2343, 2017. 7

[11] Georgia Gkioxari, Alexander Toshev, and Navdeep Jaitly. Chained predictions using convolutional neural networks. In *ECCV*, pages 728–743. Springer, 2016. 2

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 7

[13] Junjie Huang, Zheng Zhu, Feng Guo, and Guan Huang. The devil is in the details: Delving into unbiased data processing for human pose estimation. In *CVPR*, pages 5700–5709, 2020. 2, 7

[14] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *ECCV*, pages 118–134, 2018. 1, 2, 3, 5, 7, 8

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[16] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016. 2

[17] Wenbo Li, Zhicheng Wang, Binyi Yin, Qixiang Peng, Yuming Du, Tianzi Xiao, Gang Yu, Hongtao Lu, Yichen Wei, and Jian Sun. Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*, 2019. 7

[18] Ita Lifshitz, Ethan Fetaya, and Shimon Ullman. Human pose estimation using deep consensus voting. In *ECCV*, pages 246–260. Springer, 2016. 2

[19] 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*, pages 740–755. Springer, 2014. 5

[20] Buyu Liu and Vittorio Ferrari. Active learning for human pose estimation. In *ICCV*, pages 4363–4372, 2017. 2

[21] Diogo C Luvizon, Hedi Tabia, and David Picard. Human pose regression by combining indirect part detection and contextual information. *Computers & Graphics*, 85:15–22, 2019. 1, 2

[22] Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *ICML*, pages 1614–1623. PMLR, 2016. 4

[23] William McNally, Kanav Vats, Alexander Wong, and John McPhee. Evopose2d: Pushing the boundaries of 2d human pose estimation using neuroevolution. *arXiv preprint arXiv:2011.08446*, 2020. 2

[24] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499. Springer, 2016. 1, 2, 3, 7

[25] Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. Numerical coordinate regression with convolutional neural networks. *arXiv preprint arXiv:1801.07372*, 2018. 1, 2, 4, 5, 7, 8

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NIPS*, pages 8024–8035. 2019. 5

[27] Matteo Ruggero Ronchi and Pietro Perona. Benchmarking and error diagnosis in multi-instance pose estimation. In *ICCV*, pages 369–378, 2017. 2, 5

[28] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, pages 5693–5703, 2019. 1, 2, 5, 7, 8

[29] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, pages 529–545, 2018. 1, 2, 3, 5, 7, 8

[30] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *ICCV*, pages 5916–5925, 2017. 2

[31] Zhi Tian, Hao Chen, and Chunhua Shen. Directpose: Direct end-to-end multi-person pose estimation. *arXiv preprint arXiv:1911.07451*, 2019. 7

[32] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *CVPR*, pages 648–656, 2015. 7, 8

[33] Jonathan Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *arXiv preprint arXiv:1406.2984*, 2014. 2, 5

[34] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, pages 1653–1660, 2014. 2

[35] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, pages 4724–4732, 2016. 2

[36] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, pages 466–481, 2018. 1, 2, 5, 7, 8

[37] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *ICCV*, pages 1281–1290, 2017. 2

[38] Wei Yang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *CVPR*, pages 3073–3082, 2016. 2

[39] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*, pages 467–483. Springer, 2016. 2

[40] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *CVPR*, pages 7093–7102, 2020. 1, 2, 7

[41] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *ICCV*, pages 4903–4911, 2017. 5, 7, 8