DIVE DEEPER INTO INTEGRAL POSE REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Integral pose regression combines the use of an implicit heatmap with end-to-end training for human body (Sun et al., 2018; Nibali et al., 2018) and hand (Iqbal et al., 2018) pose estimation. Unlike detection-based heatmap methods, which decode final joint positions from the heatmap with a non-differentiable argmax operation, integral regression methods apply a differentiable expectation operation. The differentiable decoding allows for end-to-end training directly from ground-truth coordinates, though this slows down the learning process. It also leads to curious differences in performance, *i.e.* integral regression is competitive or better than detection-based methods on "hard" samples but perform worse on the "easy" samples (Gu et al., 2021). We do a deep dive on the inference and back-propagation of integral pose regression to better understand the causes behind the performance and training differences. For inference, we give theoretical support that expectation is better than argmax operation, but it only takes place in hard cases in practice as activation region of heatmaps shrink in easy cases. We then experimentally show that the shrinkage of activation regression is one of the main causes of its inferior performance. For back-propagation, we theoretically and empirically analyze the gradients to explain the slow training speeds. Our analysis based on expectation operation and backward propagation gives insight to understand IPR and we use experiments to demonstrate the capability of IPR to surpass the performance of detection.

1 INTRODUCTION

2D human pose estimation aims to detect the image coordinates of the body and or the hand. In the last few years, detection-based methods (Newell et al., 2016; Xiao et al., 2018; Sun et al., 2019; Li et al., 2019) and integral pose regression (IPR) methods (Sun et al., 2018; Iqbal et al., 2018) have emerged as two common paradigms for performing human pose estimation. Both detection-based and IPR methods learn a heatmap representing pixel-wise likelihoods of the joint positions; the heatmap is learned explicitly for detection-based methods, but remains implicit or "latent" for IPR. To decode the heatmaps to joint coordinates, detection methods use an argmax, while IPR methods take the expected value. As the expectation operation is differentiable, IPR has the benefit of being end-to-end learnable, even though detection methods seem more competitive accuracy-wise (COC, 2020).

Gu et al. (2021)'s recent work showed a curious performance difference between IPR and detection methods. With a nuanced split over the evaluation set, IPR outperforms detection when test samples are "harder", *i.e.* , with fewer keypoints present in the scene, under higher occlusion, and lower bounding box resolutions. Given that both detection and IPR methods work with fully convolutional feed-forward architectures, this naturally begs the question - what are the reasons behind this performance difference? Why is it that IPR, which otherwise seems to lag behind detection methods in performance (COC, 2020), can excel on these hard cases? This serves as the motivation for our closer study and analysis of integral pose regression.

Detection and IPR methods differ in two aspects (see Fig. 1 for an overview) in the forwards and backwards pass. In the forwards pass, the heatmap is decoded with an argmax for detection versus a softmax normalization and expectation for IPR. In the backwards pass, detection methods are supervised with an explicitly defined Gaussian heatmap centered on the ground truth joint, while IPR is supervised by the joint coordinates directly. Through detailed theoretical analysis and experimentation on the decoding and back-propagation process, we make following findings and contributions:

1. We propose a unified model of the heatmap to interpret and compare detection and integral regression methods. We verify the model experimentally and show that as samples progress from hard to easy, the region of activation on the heatmap decreases for both detection and integral regression methods.
2. We demonstrate experimentally that degenerately small regions of activation degrades the accuracy of both detection and integral regression methods.
3. Integral regression methods, by decoding the heatmap with an expectation operation, should result in a lower expected EPE than detection methods that decode with an argmax operation. In practice, this can be observed only for hard samples due to the shrinkage of the active region on the heatmap.
4. Direct supervision with the joint coordinates in IPR, although end-to-end, suffers from gradient vanishing and provides less spatial cues for learning the heatmap than the explicit heatmap supervision of detection methods. As a result, the training of IPR is more inefficient and slower to converge than detection methods.

In conclusion, our findings serve as insight to understand IPR which has a better decoding method theoretically and shows the potential to surpass the performance of detection by our experiments.

2 Related Work

Since the concept of "heatmap" is proposed in (Tompson et al., 2014), detection-based methods have been top performers for human pose. Existing detection-based works emphasize to extract high-quality multi-resolution features. Xiao et al. (2018) proposed to adapt ResNet with deconvolution layers while Hourglass (Newell et al., 2016) and Cascaded Pyramid Network (CPN) (Li et al., 2019) introduced cascaded network architectures with coarse-to-fine design paradigm. High-Resolution Network (HRNet) (Sun et al., 2019) follows the coarse-to-fine paradigm and further improves the performance by adapting more dense connection among different resolution representations. IPR (Sun et al., 2018) introduced a competitive regression-based framework, but exceeded by more advanced detection-based works (Xiao et al., 2018; Sun et al., 2019).

Numerical regression-based methods, which directly regress the joint coordinates, are commonly used in facial land-mark detection (Feng et al., 2018; Zhu et al., 2020). These methods however are not accurate for human pose because they fail to capture spatial knowledge. As a follow-up, IPR tries to merge the spatial knowledge from the heatmap implicitly, hence the "latent" heatmap. IPR methods are especially preferred in hand pose estimation (Spurr et al., 2020; Yang et al., 2021) but are still less common for the human pose estimation (Sun et al., 2018; Nibali et al., 2018).

Recently, two parallel lines of work have developed to further understand the behaviour of detection and IPR methods respectively. For detection-based methods, Huang et al. (2020) revealed a heatmap bias caused by the inconsistency of coordinate system transformations and solved it by redesigning the transformations during data processing. Moreover, as the predicted heatmaps during inference may violate the Gaussian assumption and worsen the performance, Huang et al. (2020) proposed to find the optimal Gaussian location based on the prediction and Zhang et al. (2020) directly modulated the predicted Gaussian distribution for decoding.

For IPR methods, Nibali et al. (2018) experimentally compared different heatmap regularizers, heatmap normalization schemes and loss functions. They find using Jensen-Shannon regularization on the heatmap with a softmax normalization and L1 loss achieves the best performance for IPR. More recently, Gu et al. (2021) performed a systematic comparison of detection vs. IPR methods with a common backbone and discovered the performance advantage of IPR on the "hard" samples. Furthermore, Gu et al. (2021) revealed a bias that arises from taking the expectation after the softmax normalization; they propose a compensation scheme to mitigate the bias which in turn improves the overall performance of IPR, making them competitive with detection methods.

The work (Zhang et al., 2020; Huang et al., 2020) focus on the explicit heatmaps of detection-based methods and investigate the bias under the Gaussian assumption. However, this is not applicable for IPR methods without the Gaussian assumption. Like (Nibali et al., 2018; Gu et al., 2021), our work also investigates the implicit heatmaps of IPR; the previous works focus on experimental exploration, while we place more emphasis on theoretical analysis to pinpoint underlying causes.

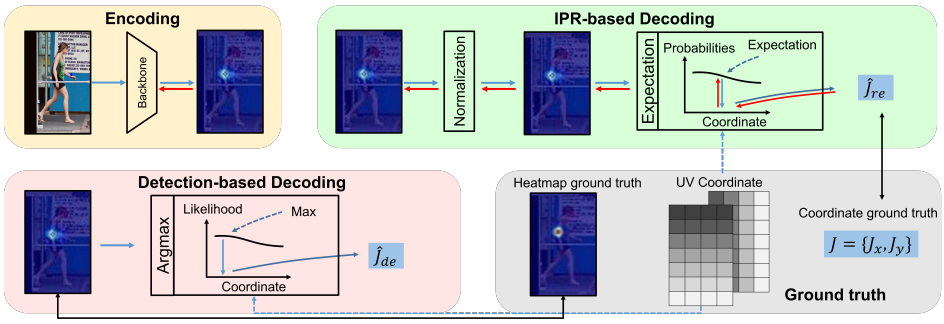


Figure 1: Comparison of the two decoding processes for detection-based methods (pink area) and IPR methods (green area). We show the encoding (yellow area) and the groundtruth (gray area), and use blue dotted line arrows for annotation. The blue/red arrows indicate the forwards/backwards pass, and the dark double arrows show the supervisions.

3 Preliminaries on Human Pose Estimation

In this work, we target the more commonly used *'top-down'* form of pose estimation in which a person detector already provides a cropped image I of the person. For simplicity, we focus our discussion on one given joint of the K total joints in the body. The pose estimation model outputs a heatmap $\hat{\mathbf{H}} \in \mathbb{R}^{M \times N}$ where M and N are the dimensions of the spatial heatmap. Typically, M and N are scaled down by a factor of 4 from the original input dimensions of I (Xiao et al., 2018; Sun et al., 2019). The heatmap $\hat{\mathbf{H}}$ represents a (discrete) spatial likelihood $P(\mathbf{J}|I)$, where $\mathbf{J} \in \mathbb{R}^{1 \times 2}$ is the 2D coordinates of that joint. In practice, all K heatmaps are predicted simultaneously by the same network, where each joint is one channel. In both detection and IPR methods, the coordinates $\hat{\mathbf{J}}$ are decoded from $\hat{\mathbf{H}}$; where the two methods differ is the manner of decoding (see Sec. 3.1) and the form of supervision applied (see Sec. 3.2).

3.1 Heatmap Decoding: Max versus Expected Value

Detection Methods apply an argmax on $\hat{\mathbf{H}}$ indexed by $\mathbf{p}$ to estimate the joint coordinates $\hat{\mathbf{J}}_{\text{de}}$:

$$\hat{\mathbf{J}}_{\text{de}} = \arg\max_{\mathbf{p}} \hat{\mathbf{H}}(\mathbf{p}), \tag{1}$$

Taking an argmax can be interpreted as taking a maximum likelihood on the heatmap $\hat{\mathbf{H}}$, assuming that $\hat{\mathbf{H}}$ is proportional to the likelihood. In practice, the final $\hat{\mathbf{J}}^{\text{de}}$ value is determined as a linear combination of the highest and second-highest response on $\hat{\mathbf{H}}$ as a way to account for quantization effects in the discrete heatmap (Newell et al., 2016). A more recent work, DARK (Zhang et al., 2020) approximates the true prediction by a Taylor series evaluated at the maximum activation of the heatmap and shows this to be more accurate.

Integral Pose Regression applies an expectation operation on $\hat{\mathbf{H}}$ to give a mean estimate of the joint coordinates. To do so, the heatmap must first be normalized to sum up to 1; the most common and effective (Nibali et al., 2018) approach is to apply a softmax normalization. Afterwards, the predicted joint $\hat{\mathbf{J}}_{\text{re}}$ with x and y components $\hat{J}_x$ and $\hat{J}_y$[1] is determined by taking the expectation on the normalized heatmap $\tilde{\mathbf{H}}$ with elements $\tilde{h}_{\mathbf{p}}$ at location $\mathbf{p}$:

$$\hat{\mathbf{J}}_{\text{re}} = \begin{bmatrix} \hat{J}_x \\ \hat{J}_y \end{bmatrix} = \sum_{\mathbf{p} \in \Omega} \mathbf{p} \cdot \tilde{h}_{\mathbf{p}} \qquad \text{where} \qquad \tilde{h}_{\mathbf{p}} = \frac{e^{\beta \hat{h}_{\mathbf{p}}}}{\sum_{(\mathbf{p}') \in \Omega} e^{\beta \hat{h}_{\mathbf{p}'}}}. \tag{2}$$

Here, Ω is the domain of the heatmap and β is a scaling parameter used in the softmax normalization. Note that softmax normalization assigns a non-zero value to *all* pixels in $\tilde{\mathbf{H}}$, even if it was originally zero in $\hat{\mathbf{H}}$. These values also contribute to the expected value, resulting in a center-biased estimated joint coordinate $\hat{J}_{\text{re}}$ (Gu et al., 2021). The smaller the β is, the stronger the bias is. Although Gu

[1]For clarity, we drop the subscript 're', as we refer to the individual components for IPR only.

et al. (2021) propose a compensation scheme, for the purposes of our analysis we will assume that β is sufficiently large that the impact of the bias is negligible.

3.2 SUPERVISION: EXPLICIT HEATMAP VS. GROUND TRUTH COORDINATES

Detection Methods are learned by providing supervision on the heatmap. The ground truth $\mathbf{H}$ is given as a circular Gaussian, with a mean centered at the ground truth joint coordinate (see Fig. 1). The loss applied is a pixel-wise MSE between the ground truth $h_{\mathbf{p}}$ and the predicted $\hat{h}_{\mathbf{p}}$:

$$L_{\text{de}} = ||\mathbf{H} - \hat{\mathbf{H}}||_2^2 = \sum_{\mathbf{p}\in\Omega} (h_{\mathbf{p}} - \hat{h}_{\mathbf{p}})^2, \tag{3}$$

where Ω is the domain of the heatmap. As the loss is defined in terms of the heatmap and not the actual variable of interest, *i.e.* the predicted joints, detection based methods are not end-to-end in their learning and this is often cited as a drawback (Sun et al., 2018; Zhang et al., 2020).

Integral Pose Regression defines a loss based on the difference between the prediction $\hat{\mathbf{J}}_{\text{re}}$ and ground truth joint location $\mathbf{J}$. The L1 loss empirically performs better than the L2 (Sun et al., 2018):

$$L_{\text{re}} = \|\mathbf{J}_{\text{gt}} - \hat{\mathbf{J}}_{\text{re}}\|_1 = (|\hat{J}_x - J_x| + |\hat{J}_y - J_y|). \tag{4}$$

IPR are end-to-end methods because they provide supervision at the joint level. As the estimated $\mathbf{H}$ is learned only implicitly, some works refer to the heatmap as "latent" (Iqbal et al., 2018).

3.3 PERFORMANCE DIFFERENCES

The performance of detection versus integral regression methods on human pose estimation has been compared by Sun et al. (2018) and Gu et al. (2021). The former found that integral regression is either as competitive or better for 2D pose estimation, though results are not fully conclusive, as Sun et al. (2018) used a different backbone for integral regression than detection.

The more standardized and detailed comparison of Gu et al. (2021) used a common backbone and partitioned the data based on three factors of variation dominating current human pose estimation benchmarks (Ruggero Ronchi & Perona, 2017): (1) the number of joints present in the scene, (2) the percentage of occlusion and (3) the largest dimension of hte bounding box input. The *"hard"* samples are those with either 1-5 joints present or $> 50\%$ occlusion or 32-64 pixel input; integral regression methods had, on average, 15% lower EPE than detection methods. The *"easy"* samples are defined by 11-17 joints, $<10\%$ occlusion and >128px input; detection methods were marginally better. The remaining cases with factors in between were considered to be *"medium"*, where performance of both methods are roughly equivalent.

4 ANALYSIS ON HEATMAP DECODING

4.1 LOCALIZED HEATMAP MODEL

To analyze the performance differences between detection and IPR methods, we make the following assumptions on the heatmap itself. First and foremost, we assume that a well-trained network will produce a heatmap $\hat{h}_{\mathbf{p}}$ with significant or large values in a *localized* support region around $\hat{\mathbf{J}}$. We denote the region of support as Φ and assume that outside of Φ, the heatmap activation is approximately zero (see Fig. 2 (a)).

Specifically, we model that the normalized heatmap[2] $\tilde{\mathbf{H}}$ is modelled by some density distribution $\mathcal{P}$. We make no assumption on the form of $\mathcal{P}$, but do assume that it fully captures the support for the joint location within Φ and that Φ is centered on the expected value of $\mathcal{P}$ on location $\boldsymbol{\mu} = (\mu_x, \mu_y)$ with an area of $(2s+1, 2s+1)$, *i.e.* ,

$$\boldsymbol{\mu} = \sum_{\mathbf{p}\in\Omega} \mathbf{p}\cdot\tilde{h}_{\mathbf{p}} = \sum_{\mathbf{p}\in\Phi(\boldsymbol{\mu},s)} \mathbf{p}\cdot\tilde{h}_{\mathbf{p}} = \left[\sum_{i=\mu_x-s}^{\mu_x+s}\sum_{j=\mu_y-s}^{\mu_y+s} i\cdot\tilde{h}_{ij}, \sum_{i=\mu_x-s}^{\mu_x+s}\sum_{j=\mu_y-s}^{\mu_y+s} j\cdot\tilde{h}_{ij}\right]^{\intercal}. \tag{5}$$

[2]For purpose of discussion, we may consider normalizing the heatmap for detection methods as per Eq. 2, which will not affect the outcome since $\arg\max(\tilde{\mathbf{H}}) = \arg\max(\hat{\mathbf{H}})$.

Experimental Verification of Localized Heatmap Model The assumption of localized activations is reasonable for detection methods as their heatmaps are learned explicitly to match a (localized) ground truth Gaussian. We empirically verify this model by statistically tallying the activations in the normalized heatmap for both detection and integral regression methods based on the easy/medium/hard splits of Sec. 3.3.

Specifically, we train both a detection and an IPR network for human pose estimation. For both networks, we use a Simple Baseline (SBL) (Xiao et al., 2018) architecture with a ResNet50 (He et al., 2016) backbone. The two networks differ only in their manner of decoding and form of supervision as outlined in Section 3. We then apply the networks on the standard human pose estimation benchmark MSCOCO (Lin et al., 2014) using the same easy, medium and hard splits as Gu et al. (2021). For each produced heatmap of 64×48, we estimate an activation sum A by summing the normalized heatmap activations within a $2s+1$ square around the ground truth coordinate, (J_x, J_y):

$$A(s) = \sum_{i=J_x-s}^{J_x+s} \sum_{j=J_y-s}^{J_y+s} \tilde{h}_{ij}. \tag{6}$$

The larger A is, the more localized the activations within the region around the ground truth joint, with a maximally achievable A of 1. Plotting A versus s (see Fig. 2 (b)), we observe that a region of 19x19, *i.e.* s=9 is sufficient to contain 95% of all non-zero activiations for both detection and regression values. Unsurprisingly, the containing region is smaller for easy samples, *i.e.* activations on the heatmap are more localized and gets progressively larger for the medium and hard samples, as the network becomes more uncertain in determining the joint location (see larger plot in Appendix A.1).

4.2 Extreme Localization as a Cause for Performance Differences

Expected End-Point-Error (EPE) For detection methods that decode the heatmap with an argmax, we assume that each position $\mathbf{p}$ has some probability $w(\boldsymbol{\mu}, s)$ of being the maximum activation, *i.e.* $\arg\max(\mathcal{P})_{\Phi(\mu,s)} \sim w(\mu, s)$. Here $w(\mu, s)$ represents a radially symmetric distribution centered on $\boldsymbol{\mu}$ with all non-zero support contained within Φ. It follows that the expected EPE for one sample for detection methods can be defined as:

$$\mathbb{E}_{\text{de}}(\boldsymbol{\mu}, s) = \sum_{\mathbf{p}\in\Phi(\boldsymbol{\mu},s)} w(\mathbf{p})||\mathbf{J}_{\text{gt}} - \mathbf{p}||_2, \tag{7}$$

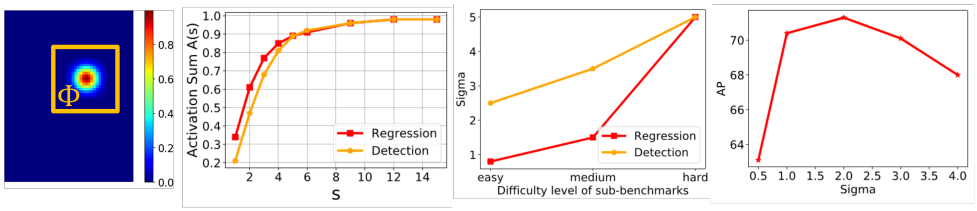
where $\mathbf{J}_{\text{gt}}$ denotes the ground truth coordinates. For integral regression methods that decode the heatmap with an expectation, the estimated joint coordinate aligns with the center of Φ by definition, *i.e.* $\hat{\mathbf{J}} = \boldsymbol{\mu}$, which leads to the following expected EPE for a single sample:

$$\mathbb{E}_{\text{re}}(\boldsymbol{\mu}) = ||\mathbf{J}_{\text{gt}} - \boldsymbol{\mu}||_2. \tag{8}$$

It can be verified (see proof in Appendix A.2) that the $\mathbb{E}_{\text{de}} \leq \mathbb{E}_{\text{re}}$. This result suggests that for some fixed $(\boldsymbol{\mu}, s)$, the expected EPE of regression should always be better than detection. However, the results of Gu et al. (2021) as summarized in Sec. 3.3 clearly show this not to be the case. This inconsistency can be better understood with the following two experiments.

Experimental Verification: IPR Methods Are "More" Localized than Detection The plot in Fig. 2 (b) indicates that regression methods seem to be *more* localized than detection methods. In particular, for smaller s, A is smaller for detection methods than integral regression methods. To verify, we compare the resulting heatmaps with an idealized Gaussian heatmap of varying standard deviations or σ, and plot the σ which gives the lowest Pearson Chi-square statistic, *i.e.* the highest similarity (see Fig. 2 (c)). The detailed similarities with different standard deviations can be seen in Appendix A.3.

As expected, the optimal σ decreases as the samples progress from hard to easy; this result is in line with Fig. 2 (b) and shows that heatmaps are more localized for easy samples. However, we also observe that the optimal σ for regression methods are much smaller for the medium and easy cases. We speculate that this difference arises due to detection methods being trained with a Gaussian heatmap of $\sigma = 2$ (Xiao et al., 2018; Sun et al., 2019). Integral regression methods, on the other hand, have no such explicit supervision and therefore no restriction on the extent of localization.



(a) Illustration of Φ (b) Activation sum $A(s)$ of $\hat{\mathbf{H}}$ (c) Optimal σ of predicted heatmaps (d) AP vs. ground truth σ

Figure 2: (a) The heatmap is only activated within Φ; (b) when s is small, regression is more squeezed; (c) from hard to esay, the σ of detection gradually decreases to 2, but that of regression has no limit; (d) the decrease of σ generally improves the performance until it's smaller than 2.

Model	Detection	Regression	$+\sigma=0.5$	$+\sigma=1$	$+\sigma=2$	$+\sigma=3$	$+\sigma=4$
AP	71.3	67.1	65.1	68.5	71.4	69.8	66.9
$\text{AP}_{e/m}$	74.3	70.4	68.2	71.9	74.1	73.0	70.2
AP_h	43.2	44.7	43.9	45.2	47.4	47.6	46.9
$A(s=2)_{e/m}$	0.49	0.65	0.75	0.63	0.54	0.49	0.45
$A(s=2)_h$	0.30	0.35	0.37	0.36	0.41	0.39	0.39

Table 1: Performance and activation sum of detection baseline, regression baseline, and regression baseline added KLDiv loss with different σs. When $\sigma=0.5, 1$, the results are not improved but even worsened. When $\sigma=2$, the spread of heatmaps are smoothed and the results even exceed detection baselines.

Experimental Verification: Extremely Localized Heatmaps Degrade Performance Is it possible that smaller hypothetical σ, *i.e.* more localized heatmaps cause poor performance? We verify this for detection methods by applying ground truth Gaussian heatmaps of varying σ for training the networks as per Sec. 4.1. We evaluate network performance using Average Precision (AP). A higher AP corresponds to more accurately located joints. Fig. 2 (d) confirms that $\sigma=2$ appears to be optimal. A smaller $\sigma = 1$ degrades the performance slightly, and there is a significant drop in the extreme case when $\sigma=0.5$. To verify the impact of σ on regression methods, we add a distribution prior to the heatmap to prevent the shrink of the support region Φ. Specifically, we apply a Kullback-Leibler Divergence (KLDiv) loss between the non-normalized heatmap $\hat{\mathbf{H}}$ and a Gaussian heatmap with varying σ. From Table 1, we see that the regression baseline is worse than detection baseline. However, when we add a prior on the heatmap distribution to encourage a sufficiently large Φ, *i.e.* ($+\sigma=1,2,3$), performance improves. At the optimal $+\sigma=2$, IPR becomes competitive with the detection baseline with the optimal and even exceeds. Most importantly, extreme values σ causes performance to degenerate, especially in the case of the extremely small $+\sigma = 0.5$. We also demonstrate the performance and localization extent of easy/medium and hard cases by AP and activation sum $A(s=2)$. After adding priors (*e.g.* , $+\sigma = 2$), $A(s=2)_{e/m}$ decreases from 0.65 to 0.54, indicating the heatmaps are less localized, and performance $\text{AP}_{e/m}$ is significantly improved. We provide the details of training of combining the two losses and magnitudes of two gradients for better understanding in Appendix A.4.

5 Analysis of Supervision and Learning

The difference in heatmap decoding not only influences the final coordinates given same assumed heatmap, but also determines the the gradients that arise from the loss. The gradients in turn influence the learning and thereby generation of the heatmaps themselves. In this section, we explore the gradients of heatmaps with respect to loss function for detection and regression based methods and pinpoint the specific gradient components which slow down the learning of integral regression.

5.1 Heatmap Gradients

Detection Methods The gradient of L_{de} (see Eq. 3) with respect to each pixel in the estimated heatmap $\hat{h}_{\mathbf{p}}$ is straightforward:

$$\frac{\partial L_{\text{de}}}{\partial \hat{h}_{\mathbf{p}}} = 2(\hat{h}_{\mathbf{p}} - h_{\mathbf{p}}). \tag{9}$$

The gradient in Eq. 9 features the predicted heatmap value $\hat{h}_{\mathbf{p}}$ and the ground truth heatmap value $h_{\mathbf{p}}$ and provides explicit supervision at every pixel. False positives *and* false negatives are penalized, by reducing wrong high likelihoods and raising incorrect low likelihoods respectively. Specifically, for each position $\mathbf{p}$ in the heatmap, if the predicted $\hat{h}_{\mathbf{p}}$ is smaller than the ground truth $h_{\mathbf{p}}$, the gradient is negative and proportionally increases the value in the next iteration and vice versa for predicted $\hat{h}_{\mathbf{p}}$ larger than the ground truth.

Integral Regression The gradient of the loss L_{re} (see Eq. 4) with respect to $\hat{h}_{\mathbf{p}}$ can be estimated based on the chain rule (see detailed derivation in Appendix A.5) as

$$\nabla_{\mathbf{p}} := \frac{\partial L_{\text{re}}}{\partial \hat{h}_{\mathbf{p}}} = \frac{\partial L_{\text{re}}}{\partial \hat{\mathbf{J}}} \frac{\partial \hat{\mathbf{J}}}{\partial \tilde{h}_{\mathbf{p}}} \frac{\partial \tilde{h}_{\mathbf{p}}}{\partial \hat{h}_{\mathbf{p}}} = \underbrace{\beta\, \tilde{h}_{\mathbf{p}}}_{\nabla_1 \text{ (value factor)}} \underbrace{(\mathrm{s}(\hat{J}_x - J_x)(i - \hat{J}_x) + \mathrm{s}(\hat{J}_y - J_y)(j - \hat{J}_y))}_{\nabla_2 \text{ (location factor)}}. \tag{10}$$

In Eq. 10, the gradient is into two terms of interest: ∇_1 and ∇_2 which we name the value factor and location factor respectively. The nature of the value and location factors give strong indications to the learning process for integral regression methods:

1. The gradient $\nabla_{\mathbf{p}}$ is proportional to the normalized predicted heatmap value $\tilde{h}_{\mathbf{p}}$ as given in the location factor. This factor makes it prone to gradient vanishing wherever locations $\mathbf{p}$ have small heatmap predictions.
2. the location factor is a linear combination of i and j, which presents a linear plane where far away points like corners are more likely to have large magnitudes.

Though these two points do not prevent the loss from decreasing during learning, they slow down the speed of training. We further elaborate on this in the following section.

5.2 A Detailed Analysis of the IPR Heatmap

We now conduct a detailed analysis of the gradients presented in Eq. 10 and show the typical cases that arise during learning. When referring to Ω, the domain of the heatmap, we use a coordinate system with the origin at the upper-left corner and increasing coordinates moving to the lower right corner (see Fig. 2 (a)). For simplicity of discussion, we assume that the ground truth joint $\mathbf{J}_{\text{gt}}$ is located somewhere in the lower-right quadrant of Ω, though the analysis holds for $\mathbf{J}_{\text{gt}}$ in all other quadrants of Ω analogously. The visualizations of the four scenarios can be viewed in the left panel of Fig. 3.

We base our analysis on the assumption that larger magnitude of $\nabla_{\mathbf{p}}$ will have larger influence on the weights of the network and effect greater change at the location $\mathbf{p}$ in the next iteration, *i.e.*

$$h_{\mathbf{p}}^{n+1} \approx h_{\mathbf{p}}^{n} - \gamma \nabla_{\mathbf{p}}^{n}, \tag{11}$$

where γ represents a step size or learning rate (see sketch of proof in Appendix A.6) and n denotes the update iteration. As the value and location factors have different contributions on the gradient from Eq. 10 and these are in turn affected by the size s and location μ of Φ, we can define four characteristic cases of Φ and outline how learning is affected.

(1) s is large, Φ has uniformly random values, *e.g.* a randomly initialized heatmap at the start of training. As $\tilde{h}_{\mathbf{p}}$ is in a similar range for all $\mathbf{p}$, the value factor ∇_1 (see Eq. 10) of all pixels are approximately similar. As such, the distinction between one pixel's gradient versus another is determined by the location factor ∇_2. Given that the ground truth coordinates (J_x, J_y) is in the lower-right quadrant, as long as (J_x, J_y) are both greater than the predicted coordinate $(\hat{J}_x, \hat{J}_y)$, $\nabla_{\mathbf{p}}$ has a gradation that increases progressively towards the bottom right corner. Generally speaking, the location factor pushes the activations of the heatmap towards the corner of the correct quadrant as corners own the largest gradients in the resultant linear plane of location factor ∇_2.

(2) s is small, μ is far from the ground truth $\mathbf{J}$, *e.g.* when μ is in the upper left quadrant of Ω while the ground truth $\mathbf{J}_{\text{gt}}$ is at the bottom right. A typical scenario of this case is if the heatmap activates around the left ankle for the ground truth right ankle. In this case, for the pixels outside of Φ, the value factor approaches zero, *i.e.* $\tilde{h}_{\mathbf{p}} \to 0$ and pushes $\nabla_{\mathbf{p}}$ towards zero and the pixels receive

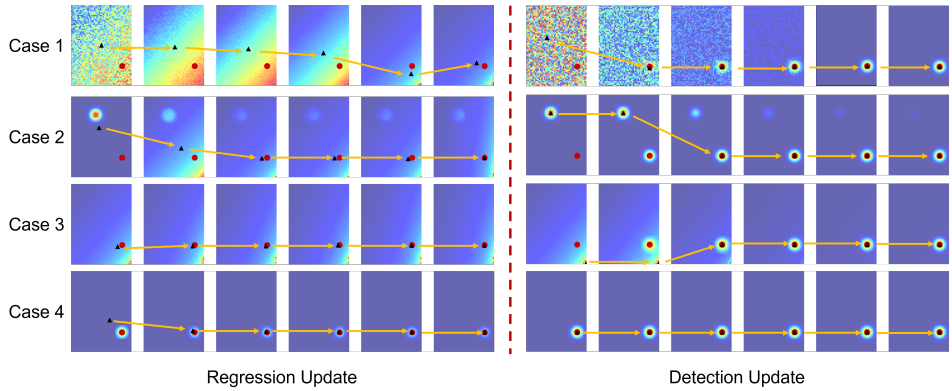


Figure 3: Toy example of regression update (left) and detection update (right). The black triangle denotes the prediction (expectation value for regression and max value for detection) while the red dot denotes the ground truth position. The arrow shows the movement of the prediction.

very limited gradient updates. For pixels in Φ, the values decrease gradually until the value factor outside of Φ no longer dominates, *i.e.* they reach the same scale, at which point the heatmap returns to case (1).

(3) s is small, μ is in the corner of the same quadrant as the ground truth $\mathbf{J}_{\text{gt}}$. Initially, this case is similar with case (2). In the process of all the elements becoming the same scale, which also means the activations moves towards the diagonal side, the prediction approaches the ground truth coordinate.

(4) s is small, μ close to ground truth $\mathbf{J}_{\text{gt}}$. This scenario occurs when the model is reasonably trained and can roughly localize the joint. As the gradient at the ground truth pixel $\mathbf{J}_{\text{gt}}$ is always negative or zero, *i.e.* ,

$$\nabla_{(J_x,J_y)} = \beta\tilde{h}_{(J_x,J_y)}(-\mathrm{s}(\hat{J}_x - J_x)(\hat{J}_x - J_x) - \mathrm{s}(\hat{J}_y - J_y)(\hat{J}_y - J_y)). \tag{12}$$

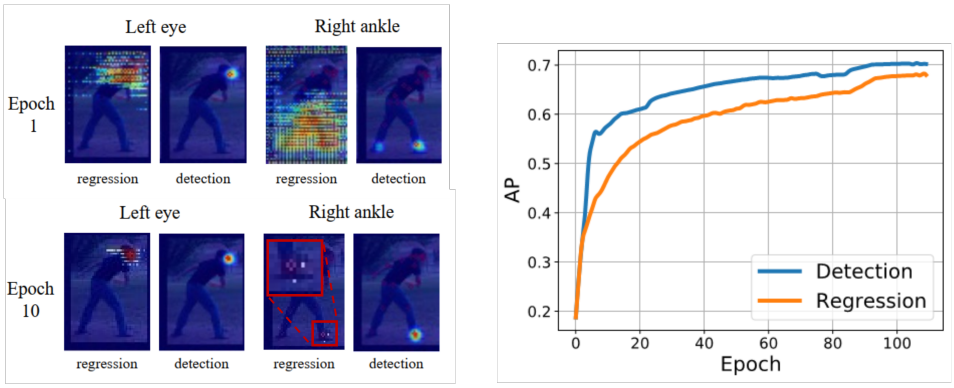
This non-positive value guides the network to predict a large heatmap response at the ground truth location, *i.e.* a large $\tilde{h}_{(J_x,J_y)}$; this in turn increases the gradient $\nabla_{(J_x,J_y)}$. This property makes the network tend to predict few extraordinary large pixels, thus shrinking the support region Φ and leads to a very small σ in Sec. 4.1.

However, for detection based methods, the ground truth heatmaps have already explicitly point out whether this pixel should be activated or not in the heatmap. Combined with its effective pixel-wise loss, the network learns faster than regression based methods. We use experiments to validate in the following section.

5.3 Experimental Verification

Idealized Sample We start by considering the case of one sample for a single joint and visualize the progression of a $(64, 48)$ heatmap as it gets updated by Eq. 11. Note that while $\nabla^n_{\mathbf{p}}$ can be calculated mathematically by Eq. 10, however we use autograd in Pytorch to obtain the gradients which we verify to be equivalent. We manually initialize the four cases from Sec. 5.2 by randomized initialization, Gaussian circle centered at some upper left point, linear plane at lower right quadrant, and Gaussian circle centered at ground truth location with $\sigma = 2$. The progression of the differently initialized heatmaps can be observed in Fig. 3.

Mathematically, it is clear from Eq. 2 that the resulting joint coordinate location from IPR can align with the ground truth joint even if the underlying heatmap does not follow a localized heatmap model. This is also illustrated in the first three cases of Sec. 5.2. However, we posit that during real-world training, over the thousands of training samples observed, over the many epochs, the network can only consistently lower the loss if the heatmap $\tilde{H}$ is learned to represent $P(\mathbf{J}|I)$. As such, the activation region Φ will be correctly localized over the corresponding semantic region in the image. Instead, what slows down convergence is the learning of Φ, or rather, the lack of direct guidance to yield a correct Φ as per detection methods. For the same four cases, we visualize the updated heatmap of Eq. 11 from the gradients defined by the detection loss in Eq. 9. Unlike regression,



(a) Comparisons of detection and regression in real training

(b) Training Efficiency of Detection vs. Regression on COCO validation set

Figure 4: (a) Comparisons of detection and regression in real training on an image selected from COCO val set at epoch 1 and 10. Results of detection have already roughly stabilized while regression can only localize in small extent in epoch 10 still with some background pixels activated. We enlarge right ankle part of regression in the red box for better visualization. More examples are shown in Appendix A.7. (b) The regression converges slower than detection method.

each case leads consistently to the same localized Φ centered on the ground truth joint coordinate, regardless of the initialization.

Real World Samples The four cases listed in Sec. 5.2 are not so clearly observable in real-world training as the output is based on the optimization results of batch-wise training on an actual network. To verify the results, we compare the training progression for a detection and IPR network under the experimental setting described in Sec. 4.1. Generally, we compare the training speeds of both detection and regression in Fig. 4 (b), which shows the slower convergence of regression method.

We visualize for one sample from the MSCOCO validation set the heatmap at epochs 1 and 10 to compare the progression of training for detection and integral regression. Fig. 4 (a) shows the heatmaps of the "left eye" and "right ankle" joints. For regression, after one epoch, the activations are widespread over a quarter to half the heatmap. This aligns roughly to a mix of Case 1 and 3 where activations are in the correct quadrant but not yet localized to a small region of support. After 10 epochs of training, predictions for the left eye (which is actually occluded, making it a "hard" sample) are localized with several activations of approximately the same value (all shown as red). For the right ankle (which is clearly present and is an "easy" sample), the network has predicts heatmap values that have collapsed spatially and dominate on one or two pixel locations.

In contrast, the detection method presents a well-localized heatmap after only one epoch of training. While there is some activation on the left ankle in epoch 1, this is unlikely to cause errors for the argmax decoding. After 10 epochs, all activations are centered around the correct right ankle.

6 Conclusion

In this paper, we dive deeper into the integral pose regression method. We explore the difference between detection and regression based methods based on the heatmap activations and gradients. Specifically, we find that the shrinkage of heatmaps of regression method is the main cause of lower accuracy. Furthermore, we investigate the gradients of IPR by giving theoretical evidence, toy problem and real world illustrations to show that the learning of IPR suffers from implicit supervisory signal. As a result, we propose a simple method based on our analysis to improve the performance of integral methods.

References

COCO Leader Board. `https://cocodataset.org/`, 2020.

Zhen-Hua Feng, Josef Kittler, Muhammad Awais, Patrik Huber, and Xiao-Jun Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In *CVPR*, pp. 2235–2245, 2018.

Kerui Gu, Linlin Yang, and Angela Yao. Removing the bias of integral pose regression. In *ICCV*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

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*, pp. 5700–5709, 2020.

Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *ECCV*, pp. 118–134, 2018.

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.

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*, pp. 740–755. Springer, 2014.

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pp. 483–499. Springer, 2016.

Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. Numerical coordinate regression with convolutional neural networks. *arXiv preprint arXiv:1801.07372*, 2018.

Matteo Ruggero Ronchi and Pietro Perona. Benchmarking and error diagnosis in multi-instance pose estimation. In *ICCV*, pp. 369–378, 2017.

Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3d hand pose estimation via biomechanical constraints. 2020.

Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, pp. 5693–5703, 2019.

Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, pp. 529–545, 2018.

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.

Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, pp. 466–481, 2018.

Linlin Yang, Shicheng Chen, and Angela Yao. Semihand: Semi-supervised hand pose estimation with consistency. In *ICCV*, 2021.

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

Beier Zhu, Chunze Lin, Quan Wang, Renjie Liao, and Chen Qian. Fast and accurate: Structure coherence component for face alignment. *arXiv preprint arXiv:2006.11697*, 2020.

A Appendix

A.1 Extension of Fig. 2 (b)

We provide a extended plot of Fig. 2 (b), which includes both detection and regression on easy, medium, and hard cases in Fig. 5. We also provide the datapoint in Table. 2.

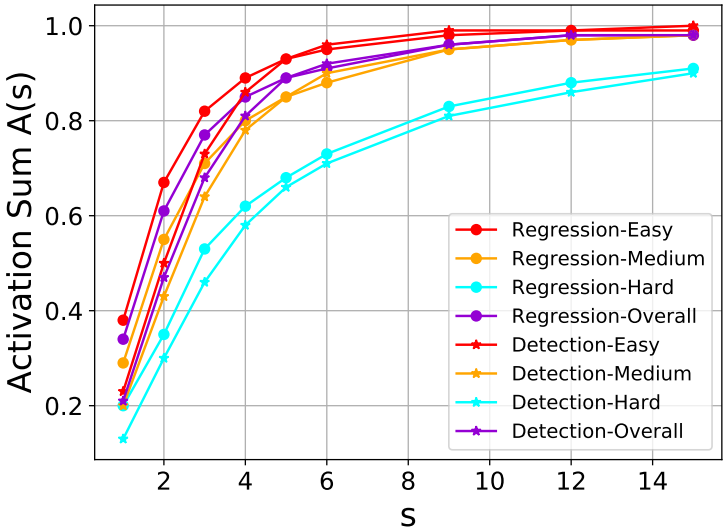


Figure 5: Activation Sum $A(s)$ of both detection and regression method on easy, medium, hard, and overall cases.

case / s	regression				detection			
	easy	medium	hard	overall	easy	medium	hard	overall
1	0.38	0.29	0.20	0.34	0.23	0.20	0.13	0.21
2	0.67	0.55	0.35	0.61	0.50	0.43	0.30	0.47
3	0.82	0.71	0.53	0.77	0.73	0.64	0.46	0.68
4	0.89	0.80	0.62	0.85	0.86	0.78	0.58	0.81
5	0.93	0.85	0.68	0.89	0.93	0.85	0.66	0.89
6	0.95	0.88	0.73	0.91	0.96	0.90	0.71	0.92
9	0.98	0.95	0.83	0.96	0.99	0.95	0.81	0.96
12	0.99	0.97	0.88	0.98	0.99	0.97	0.86	0.98
15	0.99	0.98	0.91	0.98	1.00	0.98	0.90	0.98

Table 2: Detailed datapoints to support Fig. 5.

A.2 Proof for Eq. 7⩾Eq. 8 in Sec. 4.2

In this subsection, we give detailed proof for Eq. 7>Eq. 8, which means

$$\sum_{\mathbf{p}\in\Phi(\boldsymbol{\mu},s)} w(\mathbf{p})||\mathbf{J}_{\text{gt}}-\mathbf{p}||_2 \geqslant ||\mathbf{J}_{\text{gt}}-\boldsymbol{\mu}||_2, \text{where} \sum_{\mathbf{p}\in\Phi(\boldsymbol{\mu},s)} w(\mathbf{p}) = 1. \tag{13}$$

In this way, this equation can be rewritten as

$$\sum_{\mathbf{p}\in\Phi(\boldsymbol{\mu},s)} w(\mathbf{p})||\mathbf{J}_{\text{gt}}-\mathbf{p}||_2 \geqslant \sum_{\mathbf{p}\in\Phi(\boldsymbol{\mu},s)} w(\mathbf{p})||\mathbf{J}_{\text{gt}}-\boldsymbol{\mu}||_2. \tag{14}$$

As the distribution of $\Phi(\boldsymbol{\mu}, s)$ is radially symmetric, One sufficient but unnecessary proof is that for any symmetric pair $\{\mathbf{p}_1(\mu_x - a, \mu_y - b), \mathbf{p}_2(\mu_x + a, \mu_y + b)\}$, it satisfies

$$\sqrt{(x-\mu_x-a)^2+(y-\mu_y-b)^2}+\sqrt{(x-\mu_x+a)^2+(y-\mu_y+b)^2} \geqslant 2\sqrt{(x-\mu_x)^2+(y-\mu_y)^2}. \tag{15}$$

According to Triangle Inequality in normed vector space, we arrive at

$$||\mathbf{u}||_2 + ||\mathbf{v}||_2 \geqslant ||\mathbf{u}+\mathbf{v}||_2. \tag{16}$$

When we set $\mathbf{u} = (x - \mu_x - a, y - \mu_y - b)$ and $\mathbf{v} = (x - \mu_x + a, y - \mu_y + b)$, Eq. 14 is established.

A.3 PEARSON CHI-SQUARE VALUE ON DIFFERENT CASES

We report the detailed Pearson Chi-square value between heatmaps of both detection and regression in different scenarios and Gaussian templates with different σs in Table. 3, 4, 5. We set the threshold of $A(s) > 0.8$ to show that the support region can represent the whole heatmaps. Therefore, for easy and medium case, $s=4$; while for hard cases, $s=8$.

s \ σ	regression				detection			
	2	3	4	5	2	3	4	5
8	-	4.76	4.12	3.95	-	3.51	3.31	3.25

Table 3: Results of Pearson Chi-square test between distribution of heatmap $\mathbf{H}$ and Gaussian templates $\mathbf{T}$ with different window size s and standard deviation σ in hard cases. $\sigma = 5$ is the most matched template for both detection and regression.

s \ σ	regression				detection			
	1	2	3	4	1	2	3	4
4	-	3.25	3.65	3.73	-	1.76	0.76	0.77

Table 4: Results of Pearson Chi-square test between distribution of heatmap $\mathbf{H}$ and Gaussian templates $\mathbf{T}$ with different window size s and standard deviation σ in medium cases. - denotes number larger than 1000. For detection, best matched σ should between 3 and 4; while for regression, optimal σ in medium case should between 1 and 2.

s \ σ	regression					detection					
	0.5	1	2	3	4	1	2	2.5	3	3.5	4
4	-	3.24	3.53	3.75	3.89	-	2.52	0.64	0.72	0.85	0.88

Table 5: Results of Pearson Chi-square test between distribution of heatmap $\mathbf{H}$ and Gaussian templates $\mathbf{T}$ with different window size s and standard deviation σ in easy cases. Accordingly, for detection, σ should be approximately 2.5; while for regression, optimal σ should be between 0.5 and 1.

A.4 DETAILS OF COMBINING PRIOR LOSS AND INTEGRAL LOSS

In this subsection, we will demonstrate the influence of weights of prior loss and integral loss, and the magnitudes of two gradients regarding the shared heatmaps during the training.

Let assume w_p and w_i are weights of the prior and integral losses respectively. We consider the ratio $w_p L_p : w_i \cdot \lambda L_i$ where $\lambda = 10^{-2}$ to set the two approximately equal magnitudes at the initial epoch. From Table 6, we can see that the AP heavily depends on the value of σ but for some fixed σ (within each row), the ratio of loss weightings has less impact and results are stable within $1-2\%$, with larger influence only at the extremely settings of $100:1$ or $1:100$. We also provide the ratio of gradient magnitudes in Table 7 to help explain why the weightings do not have much impact. Table 7 gives the mean ratio $\frac{g_p}{g_i}$ where g_p and g_i are the gradient magnitudes of the prior loss and integral loss respectively over the course of training. A column of 2/1/0/-1/-2 means 10^2/10^1/10^0/10^{-1}/10^{-2} ratio in gradients over the epochs. The table shows that regardless of the weighting, the gradient ratio decreases over the course of training (*i.e.* , dropoff with each column). Apart from an extremely high weighting of the prior, *i.e.* 100:1 in column 1, the prior influences training only in the initial 10 epochs. After epoch 10, the gradient ratio is around the same scale or smaller.

A.5 DERIVATION OF EQ. 10

Eq. 10 is obtained by the chain rule. Therefore, we list the value of each term as follows

$$\frac{\partial L_{\text{re}}}{\partial \hat{\mathbf{J}}} = \begin{bmatrix} s(\hat{J}_x - J_x) \\ s(\hat{J}_y - J_y) \end{bmatrix}, \quad \frac{\partial \hat{\mathbf{J}}}{\partial \tilde{h}_{\mathbf{p}}} = (i+j), \quad \frac{\partial \tilde{h}_{\mathbf{p}}}{\partial \hat{h}_{\mathbf{q}}} = \begin{cases} \beta \tilde{h}_{\mathbf{q}}(1-\tilde{h}_{\mathbf{q}}), & \mathbf{q} = \mathbf{p} \\ -\tilde{h}_{\mathbf{p}} \tilde{h}_{\mathbf{q}}, & \mathbf{q} \neq \mathbf{p} \end{cases}, \tag{17}$$

$w_p : \lambda w_i$	100:1	10:1	1:1	1:10	1:100
$+\sigma = 0.5$	63.6	64.8	65.1	65.7	66.4
$+\sigma = 1$	67.6	68.1	68.5	68.3	67.8
$+\sigma = 2$	68.5	70.8	71.4	70.6	67.8
$+\sigma = 3$	68.2	69.4	69.8	69.2	68.1
$+\sigma = 4$	64.2	66.1	66.9	66.9	67.0

Table 6: Performance of integral model adding different priors with different weight ratios $w_p : \lambda w_i$. The performance depends largely on different σs, but less influenced with reasonable change.

$w_p : \lambda w_i$	100:1	10:1	1:1	1:10	1:100
epoch 1	2	2	2	1	1
epoch 10	2	1	0	0	0
epoch 30	2	0	0	0	-1
epoch 50	1	-1	-1	-1	-1
epoch 80	0	-1	-1	-1	-2

Table 7: Ratio of gradient magnitudes of combined model (integral+$\sigma = 2$) with different weight ratios $w_p : \lambda w_i$. The gradient ratio decreases over the course of training. After epoch 10, the gradient ratio is around the same scale or smaller.

where $\mathbf{q} = (u, v)$ can be any pixel in Ω. Combining the terms, gradients at $\mathbf{p}$ can be calculated by

$$\frac{\partial L_{\text{re}}}{\partial \hat{h}_{\mathbf{p}}} = \beta((\hat{J}_x - J_x)i + (\hat{J}_y - J_y)j)\tilde{h}_{\mathbf{p}}(1 - \tilde{h}_{\mathbf{p}}) - \sum_{\substack{\mathbf{q}\in\Omega,\\ \mathbf{q}\neq\mathbf{p}}} \beta((\hat{J}_x - J_x)u + (\hat{J}_y - J_y)v)\tilde{h}_{\mathbf{q}}\tilde{h}_{\mathbf{p}}. \quad (18)$$

Eq. 10 can be obtained by rearranging the terms of Eq. 18.

A.6 Derivation of Eq. 11

Let suppose $\mathbf{H}_n = g(\mathbf{w}_n|I)$ where $\mathbf{H}_n$ is the heatmap of given image I at the n-th iteration, $g(\mathbf{w}_n)$ is the function of generating the heatmap, *i.e.* , backbone network, with weights at the n-th iteration. According to taylor expansion, we arrive at

$$g(\mathbf{w}_{n+1}) = g(\mathbf{w}_n) + \frac{g^{'}(\mathbf{w}_n)}{1!}(\mathbf{w}_{n+1} - \mathbf{w}_n) + \frac{g^{''}(\mathbf{w}_n)}{2!}(\mathbf{w}_{n+1} - \mathbf{w}_n)^2 + \ldots, \quad (19)$$

where $\mathbf{w}_{n+1} - \mathbf{w}_n = -\alpha\nabla g(\mathbf{w}) = -\alpha\frac{\partial L}{\partial \mathbf{H}}g^{'}(\mathbf{w}_n)$ according to gradient descent, and $(\mathbf{w}_{n+1} - \mathbf{w}_n)^i$ is negligible compared with $(\mathbf{w}_{n+1} - \mathbf{w}_n)$ when $i > 1$. Therefore, Eq. 19 can be rewritten as

$$\mathbf{H}_{n+1} = g(\mathbf{w}_{n+1}) \approx g(\mathbf{w}_n) + g^{'}(\mathbf{w}_n)(\mathbf{w}_{n+1} - \mathbf{w}_n) = \mathbf{H}_n - \alpha\frac{\partial L}{\partial \mathbf{H}}||g^{'}(\mathbf{w}_n)||^2. \quad (20)$$

Therefore, for each pixel $h_{\mathbf{p}}$ in $\mathbf{H}$, the update rule is

$$h_{\mathbf{p}}^{n+1} \approx h_{\mathbf{p}}^{n} - \gamma\nabla_{\mathbf{p}}^{n}, \quad (21)$$

where γ represents $\alpha\|g^{'}(\mathbf{w}_n)\|$, $h_{\mathbf{p}}^{n}$ and $\nabla_{\mathbf{p}}^{n}$ denote heatmap and gradient value located at $\mathbf{p}$ in the heatmap at nth iteration. $h_{\mathbf{p}}^{0}$ denotes the original heatmap.

A.7 Visualizations of real applications

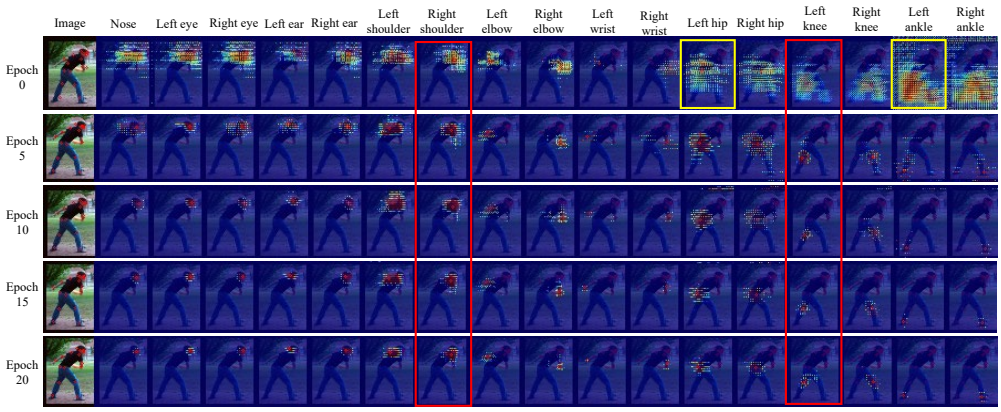


Figure 6: Visualization of training performance of integral pose regression on one single image at epoch 0, 5, 10, 15, 20 (from top to down). Corner or edge pixels of the correct quadrant are largely activated in the epoch 0 (highlighted in the yellow boxes). The further training is focused on narrowing down the activated region (shown in the red boxes).

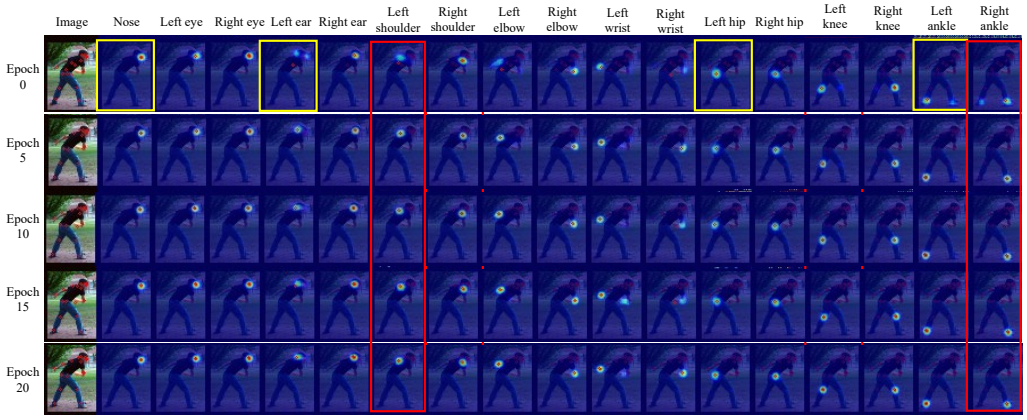


Figure 7: Visualization of training performance of detection-based methods on one single image at epoch 0, 5, 10, 15, 20 (from top to down). One epoch has already enables the network to roughly localize the joint compared with regression based method (in the yellow boxes) Further training of detection-based methods makes the predictions more stable and precise (highlighted in red boxes)