Revising Representation and Target Deviations for Accurate Human Pose Estimation

Zian Zhang, Yongqiang Zhang, Yancheng Bai, Man Zhang, Rui Tian, Yin Zhang, Mingli Ding, and Wangmeng Zuo, *Senior Member, IEEE*

***Abstract*—Owing to the normalized instance scales and robust supervision, heatmap-based human pose estimation (HPE) methods with top-down paradigm have achieved a dominant performance. However, there are two inherent deviations in the basic framework, i.e., representation and target deviations, resulting in performance bottlenecks. The representation deviation is caused by transforming various scales of instances into a unified input size, which results in performance degradation because data with different scale-related characteristics can hardly be handled via unified parameters. The target deviation is caused by exploiting a prior distribution (e.g., Gauss) to model the prediction error, which hinders sufficient network training. In this article, we propose a novel framework called DRPose to revise the above-mentioned deviations. Specifically, to address the representation deviation, a scale-aware domain bridging (SDB) block is proposed to transfer feature maps from multiple scale-dependent domains into a unified intermediate domain with dynamic parameters. To address the target deviation, a differentiable coordinate decoder (DCD) is presented to adaptively adjust target distribution of heatmaps in an end-to-end manner. Extensive experiments show that the proposed method significantly improves the performance of most existing models with negligible additional cost. Beyond this, our method achieves 77.1% AP on the COCO test-dev set, outperforming prior works with similar model complexity.**

***Index Terms*—Coordinate encoding and decoding, human pose estimation (HPE), scale-aware representation.**

I. Introduction

HUMAN pose estimation (HPE) is a fundamental problem in computer vision, since it is the basic technology for some advanced tasks such as action recognition [1], [2], [3], human motion prediction [4], [5], [6], and gait recognition [7], [8], [9]. Recently, heatmap-based HPE with top-down paradigm has achieved a leading performance due to the following two aspects.

1) *Normalized Instance Scales:* Top-down methods normalize all instances of humans into the same scale by cropping and transforming the detected human bounding boxes, and this paradigm can reduce the sensitivity of the human scale variance.
2) *Robust Supervision:* Heatmap-based method uses the Gaussian map centered on joint location as training target. That contributes to robust model fitting, because a prior noise for point estimation is provided. Furthermore, heatmap-based method supplies more dense supervision information than regressing the coordinate of joint directly, thereby enhancing the efficiency of training.

In the top-down paradigm, various scales (i.e., height and width of bounding boxes in original images) of human instances are transformed into a unified input size for batch processing. As shown in Fig. 1(a), although the sizes of original human images vary, they have similar feature distributions because all of them are sampled from real world by charge-coupled device (CCD). However, interpolating them with different scaling ratios will inevitably inject different noises into each size of images, respectively. In addition, the important information about the shape and kinetic characteristics of body, which is contained in the aspect ratio of original human instance, is lost since all images are scaled to the same size. Consequently, the preprocessing of HPE introduces a scale-related interference into the original image, which disperses the distribution domain of human images into multiple scale-dependent domains. The differences among

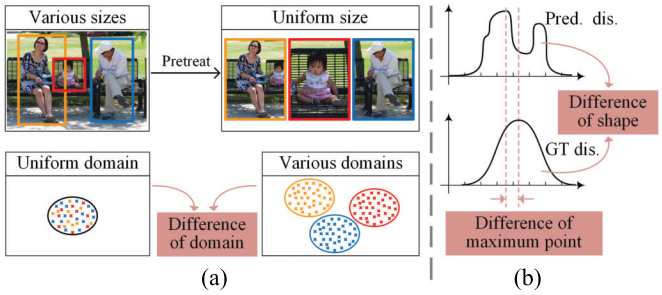

Fig. 1. Illustration of the inherent deviations in the mainstream framework of HPE. (a) Representation deviation is caused by the difference between scale-dependent data domains. (b) Target deviation is expressed by the differences of both shape and maximum point between predicted and GT distributions.

Received 27 October 2023; revised 3 October 2024, 11 February 2025, and 28 April 2025; accepted 8 May 2025. Date of publication 22 May 2025; date of current version 4 September 2025. This work was supported in part by China Postdoctoral Science Foundation under Grant 259822, in part by the National Postdoctoral Program for Innovative Talents under Grant BX20200108, in part by the National Science Foundation of China under Grant 62206077 and Grant 61976070, and in part by the Science Foundation of Heilongjiang Province under Grant LH2021F024. *(Zian Zhang and Yongqiang Zhang contributed equally to this work.) (Corresponding author: Zian Zhang.)*

Zian Zhang, Man Zhang, Rui Tian, Yin Zhang, and Mingli Ding are with the School of Instrument Science and Engineering, Harbin Institute of Technology (HIT), Harbin 150001, China (e-mail: sieann.hit@gmail.com; man.zhang.hit@gmail.com; rui.tian.hit@gmail.com; yin.zhang.hit@gmail.com; dingml@hit.edu.cn).

Yongqiang Zhang is with the College of Computer Science (College of Software-College of Artificial Intelligence), Inner Mongolia University, Hohhot 010021, China (e-mail: zhangyongqiang@imu.edu.cn).

Yancheng Bai is with the Institute of Software, Chinese Academy of Sciences, Beijing 100045, China (e-mail: yancheng.bai.1987@gmail.com).

Wangmeng Zuo is with the School of Computer Science and Technology, Harbin Institute of Technology (HIT), Harbin 150001, China (e-mail: cswmzuo@gmail.com).

Digital Object Identifier 10.1109/TNNLS.2025.3569464

scale-dependent domains degrade the model performance, as invariant parameters of convolutional neural network (CNN) struggle to adapt to diverse feature domains. In this article, we use the term "representation deviation" to define the difference in feature representation between different scale-related domains.

In addition, for heatmap-based supervision methods, the process of generating ground-truth (GT) heatmaps often involves using a manual prior distribution (e.g., Gauss) to model the prediction error. This strategy, however, introduces a target deviation, i.e., the difference between the prior GT distribution and the real distribution of the prediction error. Li et al. [10] have demonstrated that the prediction noise of HPE tends to an uneven distribution, rather than a typical Gaussian or Laplace distribution. Zhang et al. [11] have discovered that the predicted heatmap features have multiple peaks around the GT joint location. Consequently, the predicted heatmap deviates from the prior distribution in shape, and its maximum point does not align with the GT joint location due to the prediction error, as illustrated in Fig. 1(b). In this article, we use the term "target deviation" to define the difference between the predicted distribution and prior GT distribution of HPE. The target deviation results in inferior prediction accuracy due to the following reasons: 1) an erroneous joint coordinate may be obtained when decoding a predicted heatmap based on the assumption of prior distribution and 2) the predicted heatmap may not align well with the GT heatmap, thereby hindering effective network training.

Despite that the representation deviation is obvious in HPE, the problem of how to reduce the interference of the scale-dependent domain shift gains little attention. Moreover, to address the problem of target deviation, Zhang et al. [11] propose DARK to modulate the predicted heatmap into a Gauss-like distribution, and then calculate the joint coordinate by Taylor expansion. However, the distribution of predicted heatmap is still assumed to be a prior distribution in DARK. That is to say, the difference between prediction and GT distributions still exists, which hinders the further improvement of performance. Both Li et al. [10] and Du et al. [12] propose learning the true distribution of prediction errors as an implicit heatmap supervision. Although their studies demonstrate the superior effectiveness of the learned distribution over the manual distribution as a supervision signal, they fail to incorporate the strong regularization properties inherent in explicit heatmap supervision. Consequently, a comprehensive investigation into solutions for mitigating representation and target deviations becomes imperative.

In order to overcome the abovementioned problems, in this article, we propose a novel deviation-revised HPE (DRPose) framework to revise the representation and target deviations in the heatmap-based top-down HPE methods. Specifically, to address the representation deviation, we present a scale-aware domain bridging (SDB) block to transfer feature maps from multiple scale-dependent domains into a unified intermediate domain. The SDB block dynamically adapts feature maps according to the scale of human instance bounding boxes in original images, and therefore a scale-based feature normalization can be realized. Meanwhile, to revise the target deviation, we design a differentiable coordinate decoder (DCD) and a coordinate-based loss L_c to online calculate the accurate joint locations by modeling the predicted distribution of heatmaps. Significantly, benefited from the differentiability of DCD, the GT joint coordinate can be used to guide the prediction of heatmap in an end-to-end manner. Therefore, the target distribution of heatmaps will be adaptively adjusted from the prior distribution to a results-driven distribution for more abundant training. To sum-up, we make the following contributions.

1) A novel framework called DRPose is proposed to revise the representation and target deviations in the heatmap-based top-down HPE methods.
2) We propose an SDB block to resolve the problem of representation deviation caused by top-down paradigm, and this block can process various domains of features well by dynamic parameters.
3) We propose a DCD to alleviate the target deviation caused by the difference between a prior distribution (e.g., Gauss) and the real distribution of prediction error. DCD with a new designed coordinate-based loss L_c can reform the fixed coordinate decoding to a learnable operation which is end-to-end guided by the GT coordinate.
4) Extensive experiments on the COCO [13] and MPII [14] datasets demonstrate that the proposed method can improve the prediction accuracy of most existing models with negligible additional cost. Moreover, compared to advanced models, our method reaches the best performance and competitive model complexity.

II. Related Work

A. Human Pose Estimation

CNNs [15], [16], [17], [18] are widely used in the task of HPE, which greatly boosts the prediction accuracy in recent years. For the supervision manner, the methods of HPE are divided into two categories: heatmap-based methods [19], [20] and regression-based methods [21], [22]. Heatmap-based methods first predict a pixelwise heatmap, and then calculate the joint location by coordinate decoding. Regression-based methods directly regress the joint coordinate from the output of backbone. In addition, for the process flow, the methods of HPE are divided into two paradigms: top-down methods [23], [24] and bottom-up methods [25], [26]. Top-down methods first detect all persons in an image, and then perform single-person pose estimation for each human instance, respectively. Bottom-up methods directly detect all joints in an image, then associate the detected joints that belong to the same human.

Currently, heatmap-based methods with top-down paradigm have achieved a dominant performance since the normalized instance scales and the robust supervision strategy. For instance, CPN [27] uses a GlobalNet and an RefineNet to obtain precise heatmap. SimpleBaseline [28] stacks convolutional layers to enhance the estimation quality. HRNet [29] proposes a high-resolution parallel representation network to estimate joint coordinate precisely. TokenPose [30] applies transformer [31] in pose estimation to achieve competitive performance. HRFormer [32] utilizes a transformer based architecture to learn the high-resolution representation. ViTPose [33] exploits a ViT [34] architecture to achieve an excellent pose estimation accuracy. ODD [35] proposes a full-view data generation method to augment the training data, and designs an adaptive loss function to handle the foreground-background imbalance of heatmaps. HumanPoseNet [36] proposes a patch expansion and a feature refinement block to strengthen transformer-based HPE. SHaRPose [37] designs a dynamic optimization strategy for higher performance.

In this article, we focus on the heatmap-based HPE methods with the top-down paradigm, and try to further improve the pose estimation accuracy by revising two deviations, i.e., representation deviation and target deviation.

B. Scale Adaption

In the preprocessing process of the top-down paradigm, all cropped human instances are converted to a unified size. Current mainstream method can be sketched as follows: 1) modifying the aspect ratio of an original bounding box to a fixed ratio (e.g., 4:3); 2) cropping a human instance with the modified bounding box; and 3) rescaling the cropped human instance to a uniform size (e.g., 256 × 192). However, rescaling with various scaling ratios will destroy the unified distribution of original images, and modifying various aspect ratios to a fixed ratio will lose the shape and kinetic characteristics of human body contained in original bounding boxes. Despite that the representation deviation is obvious in the top-down HPE methods, the problem of how to reduce the interference of domain shift gains little attention.

Recently, for the task of scale-arbitrary super-resolution, a multiple scale-aware feature adaption method is proposed to adapt features in the backbone with a specific scale factor [38]. Specifically, a conception from conditional convolutions [39], [40], [41] is introduced to integrate the scale information with feature maps. In order to further enhance the performance of HPE, in this article, we design an SDB block to transfer feature maps from multiple scale-dependent domains into a unified intermediate domain for revising the representation deviation.

C. Target Expression

In the field of heatmap-based HPE, the problem of target expression includes coordinate encoding and decoding. During training, a GT joint coordinate is first encoded into a heatmap as the learning target. Generally, Gaussian distribution is used to model the prediction error [42] due to the prediction fuzziness. During testing, the predicted heatmap is decoded into the coordinate in the original image coordinate space. Early work [43] uses the maximum response point of heatmaps as the joint coordinate, and it is obvious that there is a quantization error in this method. To reduce the quantization error, a popular method proposed by CPN [27] adds a quarter offset in the direction from the highest response to the second highest response. G-RMI [44] first generates the joint heatmaps and offset maps simultaneously, and then aggregates them to produce the final coordinate by Hough voting. OGN [45] simplifies the aggregating process of G-RMI by selecting the maximum prediction directly. UDP [46] follows OGN and adopts a Gaussian kernel to filter the heatmap so that the highest response is located around the ground-truth label point, and generates an unbiased heatmap based on the precise floating joint coordinates instead of integer coordinates. DARK [11] assumes the predicted heatmap as a Gaussian distribution, and exploits Taylor expansion to calculate the final coordinate.

Although the above methods have achieved a significant accuracy improvement, these methods produce the final coordinates not by way of end-to-end supervision. Moreover, the difference between the prior predicted distribution and the true predicted error still exists. For the above problems, an intuitive solution is to propose an adaptively coordinate encoding and differentiable decoding methods. To this end, IPR [47] exploits a soft arg-max function to calculate the final coordinates from joint heatmaps, which enables end-to-end supervision for heatmap-based HPE. However, Gu et al. [48] demonstrate that IPR cannot effectively improve the performance of advanced HPE model, since it is easy to be disturbed by the background noise in heatmaps. RB-IPR [48] improves IPR by partitioning the heatmaps into individual regions. Nevertheless, we find that RB-IPR is still not robust enough, because the final coordinates are affected by the response values of all pixels in the selected region of heatmap. More importantly, directly supervising the training process by using the final coordinates is difficult, due to the lack of constraints on the shape of predicted distribution. PCT [49] utilizes compositional tokens as the supervision target, providing a novel way for the expression of joint. ITL [12] employs a generative model to adaptively create the target heatmap, but it lacks explicit heatmap supervision.

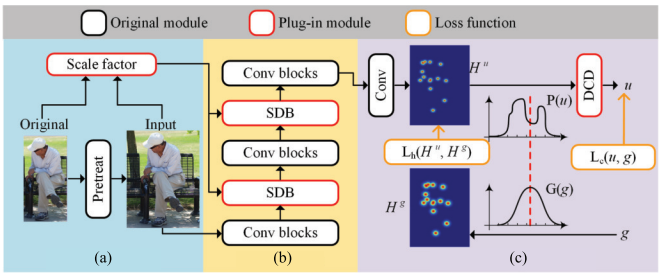

Fig. 2. Overview of our DRPose framework. The main process of predicting joint coordinates includes (a) scale factor obtained by data pretreating, (b) feature extracted by a backbone with our SDB, and (c) coordinate predicted by our DCD.

In order to achieve the purpose of adaptively tuning the target distribution while ensuring the robustness of training, in this article, we design a DCD to reform the fixed coordinate decoding to a learnable process, which is end-to-end supervised by the GT coordinate. Moreover, in the training process, our method can not only remove the interference from background pixels, but also provide constraints on the shape of the predicted distribution.

III. Proposed Method

A. Overview

Compared with other three categories of HPE methods described in Section II, the heatmap-based top-down HPE method has two merits: 1) top-down method normalizes all human instances to the same scale by cropping and transforming the detected human bounding boxes, and this paradigm can reduce the sensitivity for the scale variance of persons and 2) heatmap-based method not only provides a prior prediction noise for fitting model, but also supplies more dense supervision information than regression-based methods. However, there are two inherent deviations in heatmap-based top-down method due to the above two merits, i.e., the representation and target deviations. In this work, we focus on improving the performance of heatmap-based top-down HPE method by revising these two deviations. The overall architecture of our proposed framework, called DRPose, is shown in Fig. 2, which contains two main components (i.e., SDB and DCD) and they can be plugged into any CNN-based HPE model.

Specifically, the preprocessing phase, as shown in Fig. 2(a), resamples arbitrary-size human images into a specific unified input size, which is indispensable in top-down HPE

methods. However, resampling with various rates causes a scale-dependent domain shift of data (i.e., representation deviation). This deviation will degrade the performance since it is difficult for CNN to process features across various data domains synchronously. To overcome this deviation, we first calculate a scale factor between the original image size and the resampled image size for each image separately, then the scale factor is used to modulate the corresponding feature maps via our SDB blocks in the backbone, as shown in Fig. 2(b). The SDB block is used to adapt various domains of features into a unified intermediate domain, which will be illustrated in Section III-B in detail.

In addition, the outputted heatmap H^u expresses a distribution $\mathrm{P}(u)$ of the predicted joint coordinate u, instead of the final location. Meanwhile, most of the works model the predicted error by designing the GT heatmap H^g as a manual prior Gaussian distribution $\mathrm{G}(g)$ centered on the GT joint coordinate g. As shown in Fig. 2(c), a conventional head utilizes a 1×1 convolution to predict the H^u, and an L_2 loss function $\mathrm{L_h}$ is used to supervise training by H^g. However, there are non-negligible differences (e.g., shape and maximum point) between the $\mathrm{P}(u)$ and Gaussian distribution $\mathrm{G}(g)$ (called target deviation in this article), which also damages the prediction accuracy. In order to revise this deviation, we further design a DCD to reform the fixed postprocessing procedure of coordinate decoding to a learnable process. The DCD online calculates the final joint location u from H^u by modeling the $\mathrm{P}(u)$ instead of $\mathrm{G}(g)$. Significantly, thanks to the differentiability of DCD, g is used to supervise the prediction of heatmap by a new defined loss $\mathrm{L_c}$ in an end-to-end manner, where the training target of $\mathrm{P}(u)$ is adaptively adjusted from the prior $\mathrm{G}(g)$ to a results-driven distribution. The principle and implementation of the DCD are described in Section III-C.

B. Scale-Aware Domain Bridging

In the preprocessing phase of top-down HPE, various sizes of human images are scaled into a unified size for inputting, and such operation is usually implemented by interpolating. For the real data distribution sampled by CCD, interpolating with different scaling ratios will inject different noises into each size of images, respectively. In addition, the aspect ratio of original human instance contains significant information. We analyze the distribution of MS COCO 2017 [13] val set with different aspect ratios of bounding boxes, as shown in Fig. 3. It can be seen that the aspect ratio of original image contains the shape and kinetic characteristics of human body, e.g., smaller aspect ratios indicate standing or smaller amplitude of action and larger aspect ratios indicate horizontal posture or larger amplitude of action. That shape and kinetic characteristics of body are lost when scaling human instances to the unified size. Therefore, the preprocessing process of HPE introduces a scale-related interference into the original image, which migrates input images into a variety of scale-dependent domains. However, the fixed parameters of CNN make it difficult for dealing various domains of data, resulting in a representation deviation between the feature maps corresponding to different-sized images. That deviation limits the ability of model fitting, leading to inferior performance.

To bridge the different domains, we concatenate the aspect ratio with the scaling ratios as a scale factor, and propose a plug-in SDB block, as shown in Fig. 4, to adapt the features from various scale-dependent domains into a unified intermediate domain. The input of SDB block includes the

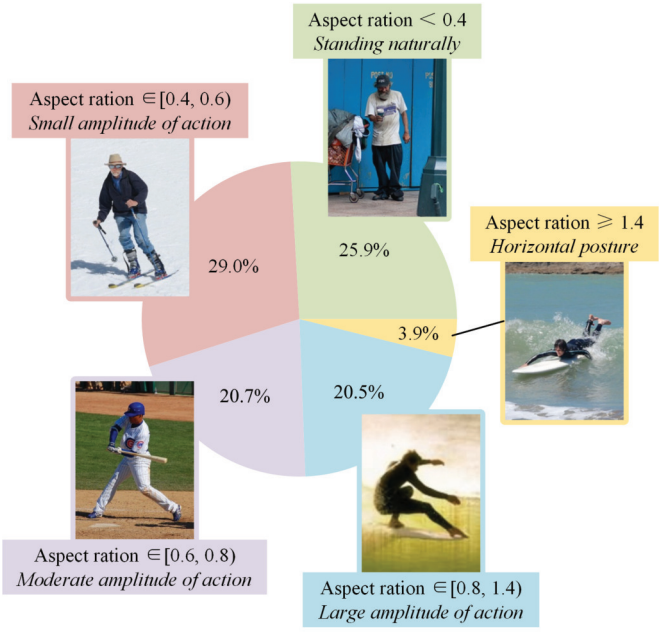

Fig. 3. Aspect ratio distribution of the COCO val set. Here, the distribution of samples based on aspect ratio is as follows: samples with an aspect ratio <0.4 (mostly naturally standing figures) account for 25.9%; samples with an aspect ratio $\in$ [0.4, 0.6) (mostly figures with small amplitude of action) account for 29.0%; samples with an aspect ratio $\in$ [0.6, 0.8) (mostly figures with moderate amplitude of action) account for 20.7%; samples with an aspect ratio $\in$ [0.8, 1.4) (mostly figures with large amplitude of action) account for 20.5%; and samples with an aspect ratio >1.4 (mostly figures in a horizontal posture) account for 3.9%.

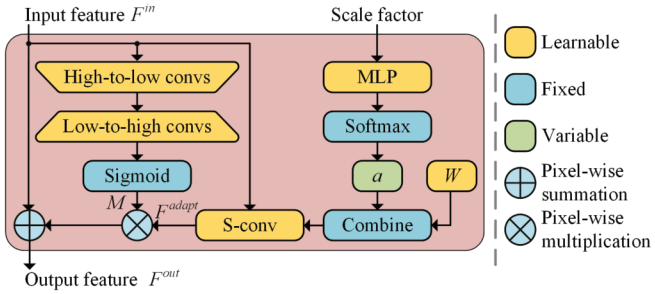

Fig. 4. Structure of our SDB block. Our SDB block is a plug-and-play feature processing module that takes scale factor as a condition.

scale factor and a feature map outputted from a convolution block in any existing backbone. Notably, consider that the structure of head network is too simple to process the adapted feature maps, SDB is designed to insert after each stage of the backbone network except the last one. The output of the SDB block is a scale-aware domain-adapted feature map, which is used as the input of the later convolutional layer.

Given a feature map F^{in}, it is first fed to a couple of high-to-low and low-to-high convolution blocks for large-region feature extraction, and then a mask map M with values ranging from 0 to 1 is generated by a Sigmoid activation function. The design of high-to-low and low-to-high convolution blocks is inspired by autoencoder [50]. This kind of networks first compresses feature map into a small-size latent by an encoder (corresponding to the high-to-low blocks), and then uses a decoder (corresponding to the low-to-high blocks) to generate an original-size feature map from the latent, resulting in superior representation capability. Moreover, F^{in} is also fed to a scale-aware convolution (i.e., the S-conv in Fig. 4) for feature

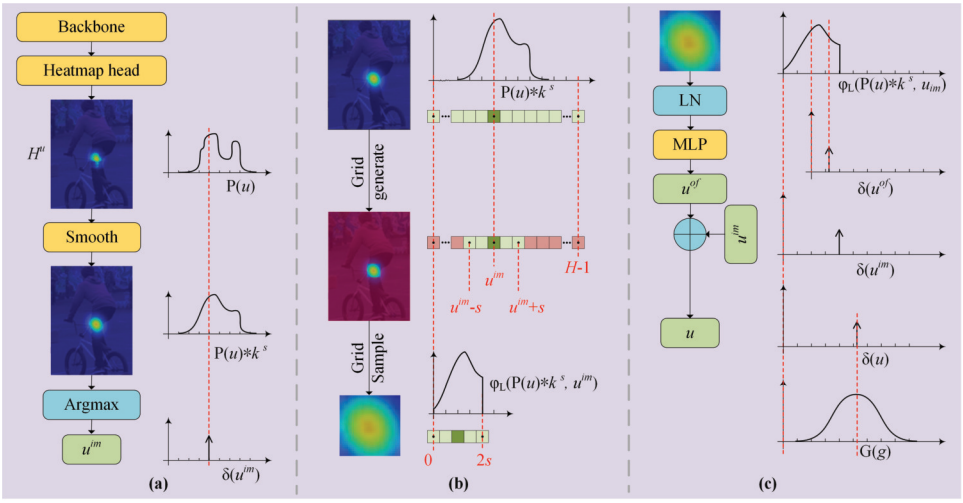

Fig. 5. Illustration of our DCD, which is performed from (a)–(c). P(·) denotes the predicted distribution, $\delta(\cdot)$ denotes the Dirac distribution, and G(·) denotes the Gaussian distribution. The entire process is briefly described as follows. First, the predicted distribution P(u) is smoothed by a low-pass filter kernel k^s to obtain a single-peak heatmap P(u) $*$ k^s. Then, an argmax function is used to calculate the integer maximum activation coordinate u^{im}, and its distribution is denoted as $\delta(u^{im})$. Next, a $(2s+1)\times(2s+1)$ sample grid centered at u^{im} is generated to sample the local area $\phi_{\mathrm{L}}(\mathrm{P(u)}*\mathrm{k^s}, u^{im})$ from the $H\times W$ heatmap P(u)$*k^s$. Finally, a neural network predicts an offset value u^{of} between u^{im} and u, so that the prediction location u can be obtained by $u = u^{im} + u^{\mathrm{of}}$.

adaption. The scale factors are fed to a multilayer perception (MLP) and a Softmax function to generate a set of coefficients a, which are used to combine with a set of learnable parameters W as the kernel of the scale-aware convolution. Here, the Softmax function is utilized to normalize the parameters of convolution kernel, and it can effectively prevent overfitting during training the whole network. Thanks to the scale-aware convolution with the generated kernel, the domain-shifted features F^{in} are adapted into a unified intermediate domain, resulting in an adapted feature map F^{adapt}. The process of F^{adapt} generation can be formulated as follows:

$$F^{\mathrm{adapt}} = (a_1 W_1 + \cdots + a_n W_n) * F^{\mathrm{in}} \tag{1}$$

where $a = \{a_1, \ldots, a_n\}$ and $W = \{W_1, \ldots, W_n\}$ are the coefficients and learnable parameters, respectively, and a and W are combined into the parameters of scale-aware convolution kernel by a linear function. In F^{in}, the domain-shift features denote the scale-variant features, but part of the features are scale-invariant. In our design, the mask map M can be used to select the scale-variant features and cover the scale-invariant features by pixelwise multiplying. Hence, the final output F^{out} of the SDB block can be formulated as follows:

$$F^{\mathrm{out}} = F^{\mathrm{in}} + F^{\mathrm{adapt}} \times M. \tag{2}$$

Intuitively, in terms of the regions with high scale-invariant across different scale factors, F^{in} can be directly used as F^{out}. In contrast, as for the regions with high scale-variant, F^{adapt} is added into F^{out} for scale-aware feature adaption. Here, the scale-aware convolution is dynamically customized based on the scale information of original images, therefore the SDB block can facilitate HPE networks to adapt various sizes of human instance images, and the effectiveness of SDB is demonstrated in Section IV-B.

C. Differentiable Coordinate Decoder

In heatmap-based HPE, the training target of a joint coordinate g is expressed by a heatmap H^g, which is a confidence map by centering a Gaussian kernel G(·) around g, and the final predicted joint coordinate is obtained by postprocessing. While heatmap-based methods achieve superior performances than regression-based methods, there is an undesirable target deviation caused by this indirect supervision. To revise the above inherent deviation, in this section, we describe the proposed DCD in detail, as shown in Fig. 5, which can reform the fixed postprocessing coordinate decoding procedure to a learnable process supervised by the GT coordinate g directly. Moreover, the differentiable DCD module with a new designed coordinate-based loss $\mathrm{L_c}$ makes the heat-map based HPE to be trained in an end-to-end manner. Note that the DCD module can not only online predict u by modeling P(u), but also adaptively adjust the training target of P(u) from the prior G(g) to a results-driven distribution.

1) Transforming and Smoothing: As shown in Fig. 5(a), the feature map outputted from the backbone is first fed to a classical head to predict a heatmap H^u in a space of $\mathbb{R}^{B\times K\times H\times W}$, where B is the batch size, K is the number of channels (i.e., the number of human joints), and H and W are the height and width of the heatmap, respectively. Then, we reshape H^u to a space of $\mathbb{R}^{BK\times 1\times H\times W}$, so that there is only one frame of heatmap H^u in a sample, which facilitates the DCD to process heatmaps. Here, H^u can be considered as a 2-D discrete distribution array, hence we use the distribution P(u) of the prediction coordinate u to denote H^u. Generally, P(u) presents multiple peaks around the maximum activation, and this may cause negative effects to the performance of our DCD. To address this issue, we exploit a learnable low-pass filter kernel k^s to smooth P(u) by a convolution operation, where a single-peak heatmap P(u)$*k^s$ is obtained.

2) Differentiable Local Sampling: To obtain the final joint location u from P(u), we crop a local area, which is a neighborhood centered on the integer maximum activation coordinate u^{im} of H^u, for accurate coordinate regression. Initially, the integer maximum activation coordinate u^{im} is calculated by executing a threshold-based argmax function on the smoothed heatmap P(u)$*k^s$, as shown in Fig. 5(a). In the process of threshold-based argmax function, we first transfer P(u)$*k^s$ to

a Dirac-like distribution $\delta_l(u^{im})$ by

$$\delta_l\left(u^{im}\right) = \frac{\mathbb{I}\left(\frac{\mathrm{P}(u)*k^s}{\max(\mathrm{P}(u)*k^s)} > \epsilon\right)}{\mathrm{Sum}\left(\mathbb{I}\left(\frac{\mathrm{P}(u)*k^s}{\max(\mathrm{P}(u)*k^s)} > \epsilon\right)\right)} \tag{3}$$

where $\epsilon \in \mathbb{R}(0,1)$ is a threshold parameter which should be as big as possible, and $\mathrm{Sum}(\cdot)$ denotes the summation of all elements. The heatmap $\mathrm{P}(u)*k^s$ is normalized by its maximum value, and the maximum activation element is selected by ϵ and set to one by an indicator function $\mathbb{I}(\cdot)$. Then, a discrete integral is utilized to calculate u^{im} by

$$u^{im} = \sum_{h=1}^{H}\sum_{w=1}^{W} \delta_l\left(u^{im}\right)_{w,h} \cdot (w,h) \tag{4}$$

where (w,h) is a coordinate in the heatmap $P(u)*k^s$. The integral process of our threshold-based argmax is similar to the soft argmax [51], but the generating of Dirac-like distribution, as shown in (3), is designed to accurately locate the coordinate of the maximum active element.

After obtaining u^{im}, all the integer coordinates in a $(2s+1)\times(2s+1)$ neighborhood centered at u^{im} are used to generate a sample grid. Then, the generated grid is utilized to crop a $(2s+1)\times(2s+1)$ local area from the $H\times W$ heatmap $\mathrm{P}(u)*k^s$ by the differentiable grid sampling [52]. The process of grid generating and grid sampling is denoted by $\phi_{\mathrm{L}}(\cdot)$, as shown in Fig. 5(b). Mathematically, the process of generating sample grid g^{im} can be described by

$$g^o = \begin{pmatrix} (-s,-s) & (1-s,-s) & \dots & (s,-s) \\ (-s,1-s) & (1-s,1-s) & \dots & (s,1-s) \\ \vdots & \vdots & \ddots & \vdots \\ (-s,s) & (1-s,s) & \dots & (s,s) \end{pmatrix} \tag{5}$$

$$g^{im} = g^o + u^{im}. \tag{6}$$

Then, we utilize U to denote the heatmap $\mathrm{P}(u)*k^s$, and use V to indicate the cropped heatmap $\phi_{\mathrm{L}}(\mathrm{P(u)}*\mathrm{k^s}, \mathrm{u^{im}})$. For $\forall i,j \in [1, 2s+1]$ and $i,j \in \mathbb{Z}$, the process of grid sampling is formulated as follows:

$$V_{(i,j)} = \sum_{h=1}^{H}\sum_{w=1}^{W} U_{(h,w)}\delta_g\left(g^{im}_{(i,j,1)}, w\right)\delta_g\left(g^{im}_{(i,j,2)}, h\right) \tag{7}$$

where

$$\delta_g\left(g^{im}_{(i,j,1)}, w\right) = \max\left(0, 1-|g^{im}_{(i,j,1)} - w|\right) \tag{8}$$

$$\delta_g\left(g^{im}_{(i,j,2)}, h\right) = \max\left(0, 1-|g^{im}_{(i,j,2)} - h|\right). \tag{9}$$

Clearly, for a set of body joint heatmaps in the $\mathbb{R}^{K\times H\times W}$ space, grid sampling involves $K\times H\times W\times(2s+1)\times(2s+1)\times 7$ floating point operations. That cost nearly matches the computational cost of a single layer of $(2s+1)\times(2s+1)$ convolution. Given that the value of s is typically small, and that grid sampling function is implemented efficiently [52]. Hence, the potential time impact of our local sampling method is minimal.

3) Location Prediction: The calculation of final coordinate is shown in Fig. 5(c). To ensure that all heatmap magnitudes are uniform, a LayerNorm [53] (LN) function is applied to normalize the cropped local area $\phi^{\mathrm{L}}(\mathrm{P}(u)*k^s)$. Then, the normalized local area is flattened to a 1-D space, and an offset value u^{of} between u^{im} and u is predicted by an MLP. Finally, u^{of} and u^{im} are added together to calculate the prediction location u. To sum-up, all the nested functions in DCD module, which predicts the final location u from the predicted distribution $\mathrm{P}(u)$, are formulated as follows:

$$u_{im} = \mathrm{argmax}_{\mathrm{thres}}\left(\mathrm{P}\left(u\right)*k^s\right) \tag{10}$$

and

$$\begin{aligned} u &= u^{im} + u^{\mathrm{of}} \\ &= u^{im} + \mathrm{MLP}\left(\mathrm{LN}\left(\phi_{\mathrm{L}}\left(\mathrm{P}\left(u\right)*k^s, u^{im}\right)\right)\right). \end{aligned} \tag{11}$$

4) Loss Function: In the DCD module, to directly predict location u, we design an $\mathrm{L_c}$ loss defined as follows:

$$\mathrm{L_c}\left(g,u\right) = \frac{1}{K}\sum_{k=1}^{K}\mathrm{SmoothL_1}\left(g_k,u_k\right)\mathbb{I}\left(|g_k-u_k|<t\right), \tag{12}$$

where K represents the number of human joints, and g_k and u_k denote the GT and predicted coordinates of joint k, respectively. Moreover, we set a threshold parameter t to exclude the predicted coordinates with large deviation by an indicator function $\mathbb{I}(\cdot)$. When supplementing $\mathrm{L_c}$ to heatmap-based supervision, the setting of threshold can effectively improve the stability of training. Overall, the training targets of DRPose include GT heatmap H^g and coordinate g, and the corresponding loss functions are $\mathrm{L_h}$ and $\mathrm{L_c}$, respectively. Our DRPose framework can be end-to-end trained by optimizing the overall objective loss function L as follows:

$$\mathrm{L}\left(H^g, H^u, g, u\right) = \mathrm{L_h}\left(H^g, H^u\right) + \mathrm{L_c}\left(g,u\right) \tag{13}$$

where $\mathrm{L_h}(H^g,H^u) = (1/K)\sum_{k=1}^{K}\mathrm{L_2}(H^g_k, H^u_k)$ supervises the prediction of heatmap H^u, and $\mathrm{L_c}$ supervises the prediction of coordinate u.

IV. Experiments

In this section, we report the performance of our proposed method in terms of accuracy and resource requirements. Moreover, we also provide experiments to verify the effectiveness of each component in our framework.

Datasets and Evaluation Metrics: We validate the proposed method by conducting experiments on MS COCO 2017 [13] and MPII [14] datasets.

MS COCO2017 contains 250 000 person samples, and each person is labeled with 17 joints. We choose the train set for training and use the val and test-dev sets for testing. For evaluation, the standard metric based on object key-point similarity (OKS) [29] is used, $\mathrm{OKS} = \left(\sum_i \exp(-d_i^2/2s_p^2k_i^2)\mathbb{I}(v_i>0)/\sum_i\mathbb{I}(v_i>0)\right)$, where d_i is the Euclidean distance between the detected joint and the corresponding ground truth, v_i is the visibility flag of the ground truth, s_p is the object scale, k_i is a per-joint constant that controls falloff, and $\mathbb{I}(\cdot)$ is an indicator function. We report the standard average precision and recall scores, i.e., AP.50 and AP.75 (AP at OKS = 0.50 and 0.75, respectively), AP (the mean of AP scores at ten positions, OKS = 0.50, 0.55,..., 0.90, 0.95), AP(M) for medium objects, AP(L) for large objects, and AR at OKS = 0.50, 0.55,..., 0.90, 0.95.

MPII contains 40 000 person samples, and each person is labeled with 16 joints. We choose the train set for training and use the val set for testing. For evaluation, the standard metric PCKh [14] score is used. A joint is correct if it falls within ηl pixels of the ground-truth position, where η is a constant and l is the head size that corresponds to 60% of the diagonal length of the GT head bounding box. We report the PCKh

TABLE I

MAIN RESULTS OF OUR PROPOSED METHOD WITH VARIOUS ARCHITECTURES AND RESOLUTIONS ON THE COCO VAL SET. WHERE † INDICATES THAT ONLY THE SDB MODULE OF THE DRPOSE FRAMEWORK IS DEPLOYED IN THE BASELINE MODEL

Method	Input size	#Params	GFLOPs	AP	AR
HRNet-W32 (Baseline) [29]	128×96	28.5M	1.8	66.9	73.7
HRNet-W32 + DRPose (Ours)	128×96	28.7M	1.8	**71.4 (+4.5)**	**77.1 (+3.4)**
SimpleBaseline-R50 (Baseline) [28]	256×192	34.0M	8.9	70.4	76.3
SimpleBaseline-R50 + DRPose (Ours)	256×192	34.1M	9.1	**72.7 (+2.3)**	**78.0 (+1.7)**
SimpleBaseline-R101 (Baseline) [28]	256×192	53.0M	12.4	71.4	77.1
SimpleBaseline-R101 + DRPose (Ours)	256×192	53.1M	12.5	**73.2 (+1.8)**	**78.6 (+1.5)**
HRNet-W32 (Baseline) [29]	256×192	28.5M	7.1	74.4	79.8
HRNet-W32 + DRPose (Ours)	256×192	28.7M	7.2	**76.5 (+2.1)**	**81.5 (+1.7)**
HRNet-W48 (Baseline) [29]	256×192	63.6M	14.6	75.1	80.4
HRNet-W48 + DRPose (Ours)	256×192	63.9M	14.7	**76.8 (+1.7)**	**81.7 (+1.3)**
HRFormer-b (Baseline) [32]	256×192	43.2M	13.3	75.6	80.8
HRFormer-b + DRPose (Ours)	256×192	43.9M	13.4	**76.7 (+1.1)**	**81.8 (+1.0)**
SimCC-W48 (Baseline) [54]	256×192	66.3M	14.6	75.9	81.2
SimCC-W48 + DRPose† (Ours)	256×192	66.6M	14.7	**76.2 (+0.3)**	81.2 (+0.0)
Poseur-W32 (Baseline) [21]	256×192	38.2M	7.9	75.3	80.4
Poseur-W32 + DRPose† (Ours)	256×192	38.3M	8.0	**75.6 (+0.3)**	**80.6 (+0.2)**
HRNet-W32 (Baseline) [29]	384×288	28.5M	16.0	75.8	81.0
HRNet-W32 + DRPose (Ours)	384×288	28.7M	16.2	**77.2 (+1.4)**	**81.9 (+0.9)**
HRNet-W48 (Baseline) [29]	384×288	63.6M	32.9	76.3	81.2
HRNet-W48 + DRPose (Ours)	384×288	63.9M	33.1	**77.3 (+1.0)**	**81.9 (+0.7)**
HRFormer-b (Baseline) [32]	384×288	43.9M	29.1	77.2	82.0
HRFormer-b + DRPose (Ours)	384×288	44.0M	29.4	**77.7 (+0.5)**	**82.6 (+0.6)**

scores of each joint, head, shoulder, elbow, wrist, hip, knee, and ankle at $\eta = 0.5$, and the mean PCKh scores at $\eta = 0.5$ and $\eta = 0.1$, respectively.

Moreover, floating point operations (FLOPs), model parameters (#Params), and running time are reported in our results to evaluate the computational efficiency of the model.

Implementation Details: We use Adam as the optimizer in our method. Unless otherwise noted, the hyperparameter ϵ in (3) is set to (1–1e^{-5}), and the hyperparameters t in (13) is set to 2. For all the used baselines, we followed the same learning schedule and epochs as in the original works. To obtain accurate GT heatmaps, we use the float joint coordinate to encode the heatmap, which is similar to DARK [11]. All the experiments are implemented by PyTorch, and are run on up to four Quadro RTX6000 GPUs.

A. Main Results

In order to verify the effectiveness of our proposed DRPose for top-down heatmap-based HPE methods, we perform experiments with various architectures (i.e., SimpleBaseline-R50 [28], SimpleBaseline-R101 [28], HRNet-W32 [29], HRNet-W48 [29], HRFormer-b [32], and SimCC-W48 [54]) and Poseur-W32 [21] and resolutions (i.e., 128 × 96, 256 × 192, and 384 × 288). The results on COCO val set are shown in Table I.

As shown in Table I, for the lightweight model HRNet-W32 with 128 × 96 resolution, our method (i.e., HRNet-W32+DRPose) outperforms HRNet-W32 by 4.5% AP and 3.4% AR, respectively. In mainstream 256 × 192 resolution, DRPose significantly improves the performance of SimpleBaseline-R50, SimpleBaseline-R101, HRNet-W32, HRNet-W48, and HRFormer-b by 2.3%/1.7%, 1.8%/1.5%, 2.1%/1.7%, 1.7%/1.3%, and 1.1%/1.0% in AP/AR, respectively. It is noteworthy that both the SimCC and Poseur model did not exploit 2-D heatmap as the training target, hence only the SDB block in our DRPose framework can be used for SimCC-W48 and Poseur-W32 baselines. Even so, a gain of 0.3% AP is achieved by our proposed method compared with SimCC and Poseur. In 384 × 288 resolution which is applied for high-precision occasions, we can see that the AP/AR of HRNet-W32, HRNet-W48, and HRFormer-b is further improved from 75.8%/81.0%, 76.3%/81.2%, and 77.2%/82.0% to 77.2%/81.9%, 77.3%/81.9%, and 77.7%/82.6% by DRPose, respectively. Moreover, as shown in Table I, DRPose not only dramatically increases the prediction accuracy of HPE models, but also the growth of execution cost of both parameters and GFLOPs are negligible (i.e., only about 0.2M and 0.2 GFLOPs additional cost). By specifically addressing representation and target deviations, our proposed DRPose delivers consistent performance gains across different architectures and resolutions, thereby confirming these deviations as fundamental challenges in HPE. Notably, our deviation mitigation approach achieves substantial accuracy gains with minimal additional computation, highlighting its practical utility.

B. Ablation Studies

To explore the effectiveness of each component in our DRPose, we perform ablation experiments and report the prediction performance on COCO val set.

1) Effect of the Scale Factor: The scale factor includes the horizontal and vertical scaling ratios and the aspect ratio of the original image in our setting, which carries the transforming information from original images to the resampled input images. Normally, the bounding boxes of human instances are detected by the pedestrian detection methods, thus the scaling ratios and aspect ratio are noisy. In the standard flow of training HPE models, the scaling ratios are multiplied with a

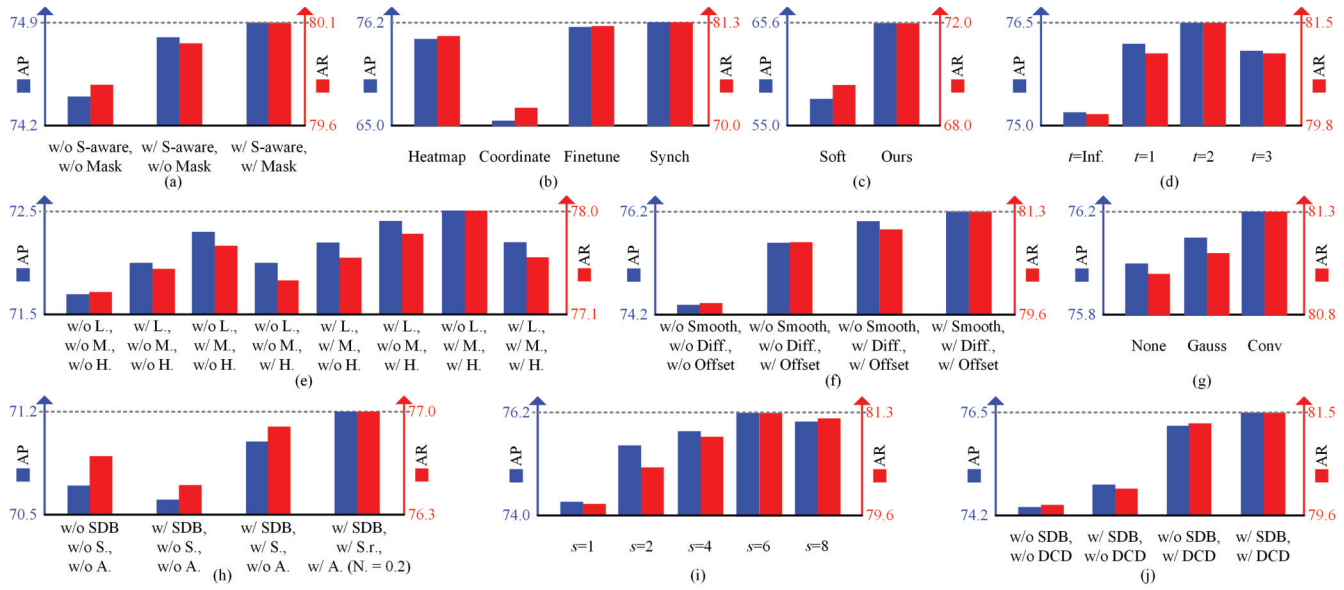

Fig. 6. Visual comparison of ablation experiments. (a) Effect of major components in SDB, corresponding to Table IV. (b) Effect of supervision strategy of DCD, corresponding to Table VII. (c) Effect of the argmax function in differentiable local sampling of DCD, corresponding to Table X. (d) Effect of threshold t, corresponding to Table XII. (e) Effect of the inserting position of SDB in backbone, corresponding to Table V, Where L. denotes Low, M. denotes Middle, and H. denotes High. (f) Effect of major components in DCD, corresponding to Table VI. (g) Effect of the smoothing filter in DCD, corresponding to Table VIII. (h) Effect of the scale factor, corresponding to Table II, Where S. denotes scaling ratios, A. denotes aspect ratio, and $N.$ denotes Neighborhood radius. (i) Effect of cropping radius s, corresponding to Table XI. (j) Effect of major components (i.e., SDB and DCD) in DRPose, corresponding to Table III.

TABLE II

EFFECT OF THE SCALE FACTOR ON THE COCO VAL SET. BASELINE: DARK-W32; INPUT SIZE: 128 × 96

SDB	Scaling ratios	Aspect ratio	Neighborhood radius	AP	AR
-	-	-	-	70.7	76.7
✓	-	-	-	70.6	76.5
✓	✓	-	-	71.0	76.9
✓	✓	✓	0	70.9	76.7
✓	✓	✓	0.1	71.0	76.9
✓	✓	✓	0.2	**71.2**	**77.0**
✓	✓	✓	0.3	71.1	76.9

TABLE III

EFFECT OF MAJOR COMPONENTS (I.E., SDB AND DCD) IN DRPOSE ON THE COCO VAL SET. BASELINE: HRNET-W32; INPUT SIZE: 256 × 192

SDB	DCD	#Params	AP	AR
-	-	28.54M	74.4	79.8
✓	-	28.70M	74.9	80.1
-	✓	28.59M	76.2	81.3
✓	✓	28.75M	**76.5**	**81.5**

random coefficient for simulating noise [44], which is an effective method for data augmentation. To apply similar noising method for the aspect ratio in training, we reset the aspect ratio by random sampling a value from its neighborhood, and the radius of neighborhood is set as a hyperparameter. In order to explore the effectiveness of our design for the scale factor, ablation experiments are conducted based on DARK-W32 baseline with 128 × 96 resolution on COCO dataset.

As shown in the first and the second rows of Table II, when plugging the SDB block into network but setting the scale factor as a constant, the accuracy of baseline is reduced to 70.6%/76.5% in AP/AR. This phenomenon proves that simply adding several convolutional layers (i.e., SDB block) to the backbone without considering the scale factor cannot improve the accuracy, i.e., the scale factor plays an important role in our framework. Then, when only using scaling ratios in SDB, the AP/AR performance is increased from 70.7%/76.7% to 71.0%/76.9%. After adding the aspect ratio without noising (i.e., radius of neighborhood =0), the performance is decreased to 70.9%/76.7% from 71.0%/76.9% in AP/AR. This result shows that the noise of aspect ratio has hindered model fitting. To this end, we conduct data augmentation for the aspect ratio, and the AP/AR is improved to 71.0%/76.9%, 71.2%/77.0%, and 71.1%/76.9% with 0.1, 0.2, and 0.3 neighborhood radius, respectively. We visually compare these results in Fig. 6(h), it can be seen that using scaling ratios as the scale factor is effective, and adding moderate noising (e.g., 0.2 neighborhood radius) into the aspect factor can further enhance the detection performance.

2) Effect of Major Components in DRPose: In order to verify the effectiveness of major components (i.e., SDB block and DCD) in DRPose, we ablate each component individually and conduct experiments with HRNet-W32 baseline under the resolution of 256 × 192. As shown in Table III, the SDB block improves AP/AR of baseline from 74.4%/79.8% to 74.9%/80.1% with a moderate cost of extra resource consumption. As mentioned earlier, the reason is that all the feature maps are transferred to a unified domain by the SDB block, which is helpful for the feature extraction of HPE models. Moreover, the DCD brings the gains of 1.8% AP (from 74.4% to 76.2%) and 1.5% AR (from 79.8% to 81.3%) with a marginal extra cost of memory cost. These large margin improvements prove that the DCD can adaptively modulate training target into a more effective distribution. Finally, two components in our framework are complementary, and using them together (i.e., DRPose) obtains 76.5% AP and 81.5% AR with 2.1%/1.7% improvement in AP/AR. Furthermore, the visual comparison illustrated in Fig. 6(j) clearly validates the claim that revising either representation or target deviation is helpful for more accurate joint location.

TABLE IV

EFFECT OF MAJOR COMPONENTS IN SDB ON THE COCO VAL SET. BASELINE: HRNET-W32; INPUT SIZE: 256 × 192

S-aware	Mask	#Params	AP	AR
-	-	28.53M	74.4	79.8
✓	-	28.54M	74.8	80.0
✓	✓	28.70M	**74.9**	**80.1**

TABLE V

EFFECT OF THE INSERTING POSITION OF SDB IN BACKBONE ON THE COCO VAL SET. BASELINE: DARK-R50; INPUT SIZE: 256 × 192

Num	Low	Middle	High	#Params	AP	AR
(a)	-	-	-	34.05M	71.7	77.3
(b)	✓	-	-	34.06M	72.0	77.5
(c)	-	✓	-	34.09M	72.3	77.7
(d)	-	-	✓	34.13M	72.0	77.4
(e)	✓	✓	-	34.11M	72.2	77.6
(f)	✓	-	✓	34.14M	72.4	77.8
(g)	-	✓	✓	34.17M	**72.5**	**78.0**
(h)	✓	✓	✓	34.18M	72.2	77.6

TABLE VI

EFFECT OF MAJOR COMPONENTS IN DCD ON THE COCO VAL SET. BASELINE: HRNET-W32; INPUT SIZE: 256 × 192

Smooth	Diff.	Offset	#Params	AP	AR
-	-	-	28.54M	74.4	79.8
-	-	✓	28.58M	75.6	80.8
-	✓	✓	28.59M	76.0	81.0
✓	✓	✓	28.59M	**76.2**	**81.3**

TABLE VII

EFFECT OF SUPERVISION STRATEGY OF DCD ON THE COCO VAL SET. BASELINE: HRNET-W32; INPUT SIZE: 256 × 192

Supervision strategy	DCD	L_h	L_c	#Params	AP	AR
Heatmap (baseline)	-	✓	-	28.54M	74.4	79.8
Coordinate	✓	-	✓	28.59M	65.6	72.0
Finetune	✓	✓	✓	28.59M	75.7	80.9
Synch	✓	✓	✓	28.59M	**76.2**	**81.3**

TABLE VIII

EFFECT OF THE SMOOTHING FILTER IN DCD ON THE COCO VAL SET. METHOD: HRNET-W32+ DCD; INPUT SIZE: 256 × 192

LPF	GFLOPs	AP	AR
None	7.12	76.0	81.0
Gaussian	7.12	76.1	81.1
Conv	7.12	**76.2**	**81.3**

TABLE IX

EFFECT OF HYPERPARAMETER ϵ ON THE COCO VAL SET. METHOD: HRNET-W32+ SDB + DCD; INPUT SIZE: 256 × 192

ϵ	0.3	0.6	0.9	1-1e-5	1-1e-9
RMSE	8.5	8.5	6.3e-1	**2.9e-4**	nan

TABLE X

EFFECT OF THE ARGMAX FUNCTION IN DIFFERENTIABLE LOCAL SAMPLING OF DCD ON THE COCO VAL SET. METHOD: HRNET-W32+ DCD; INPUT SIZE: 256 × 192

Argmax	GFLOPs	AP	AR
Soft	7.12	57.8	69.6
Our	7.12	**65.6**	**72.0**

TABLE XI

EFFECT OF CROPPING RADIUS s ON THE COCO VAL SET. METHOD: HRNET-W32+ DCD; INPUT SIZE: 256 × 192

s	1	2	4	6	8
AP	74.3	75.5	75.8	**76.2**	76.0
AR	79.8	80.4	80.9	**81.3**	81.2

TABLE XII

EFFECT OF THRESHOLD T ON THE COCO VAL SET. METHOD: HRNET-W32+ DRPOSE; INPUT SIZE: 256 × 192

t	Inf.	1	2	3
AP	75.2	76.2	**76.5**	76.1
AR	80.0	81.0	**81.5**	81.0

The combination of the two components achieves the highest performance improvement as their optimization targets are distinct. Specifically, the SDB block enhances the scale-related feature representation in the feature maps, while the DCD optimizes the decoding process for heatmaps. Therefore, the integration of these two components into other frameworks is flexible and does not require a specific order.

3) Effect of Major Components in SDB: The SDB block includes a scale-aware module (labeled as "S-aware") used to generate adapted feature map F^{adapt}, and a mask module (labeled as "Mask") used to generate the mask map M. To verify the effectiveness of these modules, we ablate each module individually and the results are shown in both Table IV and Fig. 6(a). When only using "S-aware," the AP/AR of baseline is increased from 74.4%/79.8% to 74.8%/80.0% with 0.01M extra cost of memory footprint. On this basis, adding "Mask" to SDB block improved by 0.1%/0.1% in AP/AR, and the parameter increment is 0.16M (28.54M versus 28.70M). Therefore, all components in our SDB block are effective.

4) Exploring the Compatibility of SDB and Backbone: Our SDB is a plug-in feature processing module, hence the model performance is influenced by the inserted position of SDB in the backbone. We divide a backbone into three parts: the first one-third of network extracts the low-level image information, the middle-level feature maps contain roughly location information of joints, and the last one-third of backbone generates the high-accuracy location map. Based on the above setting, we conduct experiments by inserting the SDB blocks into different positions in a DARK-R50 model, and the results on COCO val set are reported in Table V. As shown in Table V(b)-(d), inserting a SDB block into the low, middle, and high level of feature processing module can improve the AP/AR of baseline from 71.7%/77.3% to 72.0%/77.5%, 72.3%/77.7%, and 72.0%/77.4%, respectively. The above results prove that inserting the SDB block into any stage of backbone can present a good compatibility, and the greatest gain is obtain by processing middle-level features. Table V(e)-(g) shows that transferring middle and high-level feature maps to a uniform domain achieves the best prediction performance (i.e., 72.5%/78.0% AP/AR). Nevertheless, as shown in Table V(h), when using scale-aware domain adaption in all the stages of backbone simultaneously, the improvement is inferior. We think the reason is that more levels of domain adaption bring more burdens for network training. To sum-up, as illustrated in Fig. 6(e), inserting the SDB blocks in middle and high-level stages of backbone is considered as the most compatible setting, and we use this setting in our other experiments.

TABLE XIII
INCREMENT OF COMPUTATIONAL RESOURCES IN DIVERSE SCENARIOS, WHERE "R." DENOTES RUNNING TIME

Base model	Input size	DRPose	#Para (M)	GFLOPs	R. (ms)
HRNet-W32	128×96	-	28.54	1.78	5.0
HRNet-W32	128×96	✓	28.69 (+0.15)	1.80 (+0.02)	5.4 (+0.4)
HRNet-W32	256×192	-	28.54	7.12	6.5
HRNet-W32	256×192	✓	28.71 (+0.17)	7.20 (+0.08)	7.2 (+0.7)
HRNet-W32	384×288	-	28.54	16.02	10.8
HRNet-W32	384×288	✓	28.73 (+0.19)	16.20 (+0.18)	12.5 (+1.7)
HRNet-W48	128×96	-	63.60	3.65	5.2
HRNet-W48	128×96	✓	63.79 (+0.19)	3.68 (+0.03)	5.8 (+0.6)
HRNet-W48	256×192	-	63.60	14.61	7.8
HRNet-W48	256×192	✓	63.80 (+0.20)	14.73 (+0.12)	9.0 (+1.2)
HRNet-W48	384×288	-	63.60	32.88	13.3
HRNet-W48	384×288	✓	63.83 (+0.23)	33.14 (+0.26)	16.1 (+2.8)

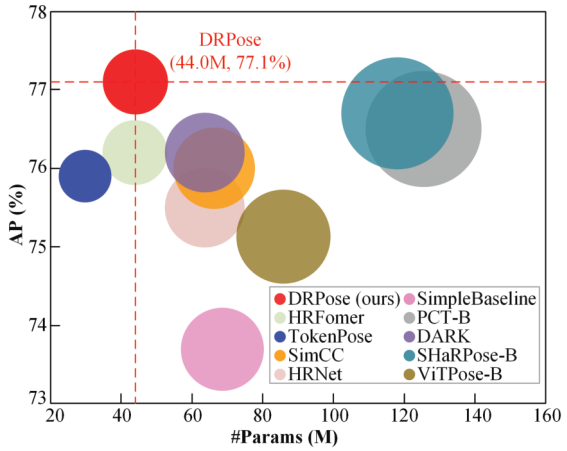

Fig. 7. Visual comparison of our method with some SOTA models on the COCO test-dev set. The *y*-axis is AP, the *x*-axis and bubble size are the number of parameters. Our method achieve best performance with competitive number of model parameters. Best be viewed in color and zoomed in.

5) Effect of Major Components in DCD: The key function of our DCD includes regressing offset from local heatmap (labeled as "Offset"), differentiable decoding (labeled as "Diff."), and smoothing (labeled as "Smooth"). To verify the effectiveness of these designs, we ablate each module individually and the results are shown in Table VI. When only using "Offset," the AP/AR of baseline is increased from 74.4%/79.8% to 75.6%/80.8% AP/AR with 0.04M extra cost of memory footprint. On this basis, adding "Diff." to DCD can improve the score to 76.0%/81.0% AP/AR, and the parameter increment is 0.01M. Then using "Smooth" can further enhance the performance to 76.2%/81.3% AP/AR with negligible extra cost (<0.01M). A more intuitive visual analysis is shown in Fig. 6(f). It can be demonstrated that regressing offset from local heatmap is the most important part of DCD, and the differentiable design and smoothing is also valuable. Therefore, we include all these components in our final DCD model.

6) Effect of the Supervision Strategy in DCD: To analyze the effectiveness of the supervision strategy in DCD module, several experiments are conducted based on the HRNet-W32 baseline with 256 × 192 resolution under different training strategies, and the results are shown in Table VII and Fig. 6(b). First, we train the baseline with DCD module supervised only by the coordinate loss L_c. Although the prediction accuracy is inferior (65.6%/72.0% AP/AR) compared with the baseline (74.4%/79.8% AP/AR), it proves that our proposed DCD with L_c can enable the HPE network to be trained in an end-to-end way. To further improve the estimation accuracy, we concatenate the DCD to a parameters-fixed HRNet-W32 which is well trained with L_h, and then merely train the DCD, denoted as "Finetune." This setting increases the performance of baseline by 1.3%/1.1% AP/AR absolutely, which demonstrates that the DCD with L_c can precisely model the predicted distribution $P(u)$ in heatmap H^u and calculate the accurate joint coordinate u. Finally, when utilizing both heatmap loss L_h and coordinate loss L_c to end-to-end train the whole network synchronously (denoted as "Synch"), the prediction accuracy is further improved to 76.2%/81.3% AP/AR, which verifies that the model has been optimized by adjusting the training target of $P(u)$.

7) Effect of the Smoothing Filter in DCD: The effectiveness of smoothing filter in the DCD is validated on the COCO val set, as shown in Table VIII and Fig. 6(g). When utilizing a Gaussian kernel to smooth heatmaps, the model performance is improved from 76.0%/81.0% to 76.1%/81.1% in AP/AR compared to model without LPF filter. Furthermore, replacing the Gaussian kernel with a learnable convolution kernel further achieves a gain of 0.1%/0.2% in AP/AR (from 76.1%/81.1% to 76.2%/81.3%). Therefore, the adaptive smoothing filter is valuable for predicting coordinates in the DCD.

8) Effect of the Argmax Function in DCD: To select a reasonable value of hyperparameter ϵ for our threshold-based argmax in differentiable local sampling of DCD, we conduct experiments on the COCO val set under various ϵ [refer to (3)]. Here, the root mean squared error (RMSE) is used to quantify the mean positioning error of the maximum elements outputted by the threshold-based argmax function. Theoretically, the value of ϵ is closer to 1, the positioning error of our threshold-based argmax is smaller. However, as shown in Table IX, when setting ϵ to $1–1e^{-9}$, the out-of-memory exception occurs due to the finite word length. Hence, we set ϵ as a moderate value (i.e., $1–1e^{-5}$) for effectively positioning maximum element.

In addition, to compare the effect of our threshold-based argmax with the existing soft argmax, we train two HRNet-W32+ DCD models supervised by only joint coordinate, which calculates the maximum element coordinates by the above two methods, respectively. As shown in Table X and Fig. 6(c), compared to the soft argmax, a superior model performance (65.6%/72.0% AP/AR) is achieved by our proposed threshold-based argmax with the same cost, and we think the reason is that our method is more suitable for indexing the integer coordinate of maximum elements.

9) Effect of the Cropping Radius s in DCD: The cropped area is crucial for predicting offsets in DCD, as a smaller area may not offer sufficient information, while an excessively large area can introduce unnecessary noise. To select an appropriate cropped area, we conduct experiment to analyze the effect of the cropping radius s in DCD, and the results are shown in Table XI and Fig. 6(i). In the case of 256 × 192 input size, using a proper cropping radius (i.e., $s = 6$) achieves best accuracy (i.e., 76.5%/81.3% in AP/AR). Significantly, in heatmap-based paradigm, the scale of the joint response region is linearly proportional to the scale of the heatmap. Therefore, the cropping radius s is set to 3 for an input size of 128 × 96, and to 9 for an input size of 384 × 288.

TABLE XIV

COMPARISON WITH THE STATE-OF-THE-ART HPE METHODS ON THE COCO TEST-DEV SET, WHERE ⋄ DENOTES THE MODEL IS TRAINED BY OURSELVES WITH THE SAME CONFIGURATIONS AS OFFICIAL REPORT

Method	Backbone	Input size	GFLOPs	#Params	AP	AP.50	AP.75	AP(M)	AP(L)	AR
SimpleBaseline [28]	ResNet-152	384×288	35.6	68.6M	73.7	91.9	81.1	70.3	80.0	79.0
HRNet [29]	HRNet-W48	384×288	32.9	63.6M	75.5	92.5	83.3	71.9	81.5	80.5
DARK [11]	HRNet-W48	384×288	32.9	63.6M	76.2	92.5	83.6	72.5	82.4	81.1
TokenPose [30]	TokenPose-L/D24	384×288	22.1	29.8M	75.9	92.3	83.4	72.2	82.1	80.8
HRFormer [32]	HRFormer-b	384×288	29.1	43.9M	76.2	92.7	83.8	72.5	82.3	81.2
SimCC [54]	HRNet-W48	384×288	32.9	66.3M	76.0	92.4	83.5	72.5	81.9	81.1
ViTPose-B [33]	ViT-B	256×192	16.8	85.9M	75.1	92.5	83.1	72.0	80.7	80.3
ViTPose-L$^{\diamond}$ [33]	ViT-L	256×192	59.8	308.5M	75.9	92.6	83.6	72.6	81.5	81.0
ODD [35]	HRNet-W32	384×288	16.0	28.5M	75.9	92.5	83.5	72.4	81.8	-
PCT-B [49]	Swin-B	256×256	15.2	125.5M	76.5	92.5	84.7	-	-	-
SHaRPose-B [37]	SHaRPose-B	384×288	32.9	118.1M	76.7	92.8	84.4	73.2	82.6	81.6
DRPose (Ours)	HRFormer-b	384×288	29.4	44.0M	**77.1**	**92.8**	**84.5**	**73.4**	**83.2**	**81.9**

Fig. 8. Qualitative evaluation results for joints detection on the COCO val set. The red and green skeletons denote the estimated results of the baseline and our DRPose, respectively. Baseline: SimpleBaseline-R50; Input size: 256 × 192. Best be viewed in color and zoomed in.

10) Effect of the Loss Threshold t: The effectiveness of threshold t in loss function L_c [see (12)] is validated by the results in Table XII and Fig. 6(d). When canceling the threshold (i.e., t = infinite), the model performance is 75.2%/80.0% in AP/AR. In contrast, a proper threshold (i.e., t = 2) enhances

TABLE XV
COMPARISON WITH THE STATE-OF-THE-ART HPE METHODS ON THE MPII VAL SET, WHERE ⋄ DENOTES THE MODEL IS TRAINED BY OURSELVES WITH THE SAME CONFIGURATIONS AS OFFICIAL REPORT

Method	Backbone	GFLOPs	Params	Hea	Sho	Elb	Wri	Hip	Kne	Ank	PCKh@0.5	PCKh@0.1
SimpleBaseline [28]	ResNet-152	20.9	68.6M	97.0	95.9	90.0	85.0	89.2	85.3	81.3	89.6	-
HRNet [29]	HRNet-W32	9.5	28.5M	97.1	95.9	90.3	86.5	89.1	87.1	83.3	90.3	37.7
DARK [11]	HRNet-W32	9.5	28.5M	97.2	95.9	91.2	86.7	89.7	86.7	84.0	90.6	42.0
TokenPose [30]	TokenPose-L/D24	13.8	28.1M	97.1	95.9	90.4	86.0	89.3	87.1	82.5	90.2	-
SimCC [54]	HRNet-W32	9.5	32.7M	97.2	96.0	90.4	85.6	89.5	85.8	81.8	90.0	36.8
HDM [55]	HRNet-W32	-	-	97.3	96.2	91.2	86.8	90.1	87.4	84.1	90.9	-
PCT-B$^{\diamond}$ [49]	Swin-B	15.2	125.5M	95.6	95.7	89.6	83.0	88.7	83.6	75.9	88.2	28.5
HumanPoseNet-B [36]	DaViT-B	-	-	97.1	**96.6**	91.5	**88.0**	**90.8**	88.4	84.6	91.4	-
SHaRPose-B [37]	SHaRPose-B	-	-	-	-	-	-	-	-	-	91.4	-
DRPose (Ours)	HRNet-W32	9.6	28.7M	**97.4**	96.2	**91.9**	87.9	90.4	**88.7**	**85.5**	**91.5**	**43.6**

the prediction accuracy to 76.5%/81.5% in AP/AR, so that we set $t = 2$ in our experiments.

11) Quantitative Analysis for Computational Resources: As previously discussed, DRPose will slightly increase the computational resources requirement. To investigate the specific impact, we conduct a quantitative analysis of the increases in computational resources brought by DRPose under different model scales and input resolutions. The base models include HRNet-W32 and HRNet-W48, and the resolutions are 128×96, 256×192, and 384×288, respectively. Our experimental results are shown in Table XIII, the increments in model parameters, GFLOPs, and running time are reported. Regarding the model scale, the additional computational resources required by DRPose increase with the model scale. For example, when introducing DRPose on HRNet-W32 (256×192), the additional computational resources are +0.17M parameters and +0.08 GFLOPs and +0.7 ms running time, while they increase to +0.20M parameters and +0.12 GFLOPs and +1.2 ms running time when DRPose is introduced on HRNet-W48 (256×192). This is because larger scale models contain more feature layer channels, which requires the SDB block to configure more convolutional parameters. Moreover, we can see that the additional computational resources required by DRPose increase with input resolution. For example, when introducing DRPose on HRNet-W32 (128×96), the additional computational resources are +0.15M parameters and +0.02 GFLOPs and +0.4 ms running time, while they increase to +0.19M parameters and +0.18 GFLOPs and +1.7 ms running time when DRPose is introduced on HRNet-W32 (384×288). The reasons are twofold: 1) increasing the input resolution enlarges the feature maps, thereby requiring the SDB block to perform more computations and 2) increasing the input resolution also increases the resolution of the predicted heatmaps, thereby necessitating the DCD to crop larger regions for prediction. Accordingly, we can conclude that the additional computational resources introduced by DRPose are influenced by the model scale and input resolution, with larger models or higher resolutions demanding more additional computational resources. However, as shown in Table XIII, the additional computational resources introduced by DRPose are acceptable, thus indicating its practical value.

C. Comparison With State-of-the-Arts

We compare our DRPose with SOTA methods on COCO test-dev set, as shown in Table XIV. We observe the following: 1) our method achieves the best performance (77.1%/81.9% AP/AR) with competitive efficiency, i.e., 44.0M parameters; 2) DRPose with HRFormer-b exceeds the newest model SHaRPose-B by 0.4%/0.3% in AP/AR absolutely, and the number of model parameters is decreased from 118.1M to 44.0M; and 3) compared to our baseline model HRFormer-b, DRPose achieves the gains of 0.9%/0.7% (from 76.2%/81.2% to 77.1%/81.9%) AP/AR, with minimal extra execution cost, i.e., 0.1M parameters. Furthermore, we represent key attributes (i.e., AP and number of model parameters) of these SOTA methods in a bubble chart form, as shown in Fig. 7. Clearly, our method reaches the best performance with competitive model complexity.

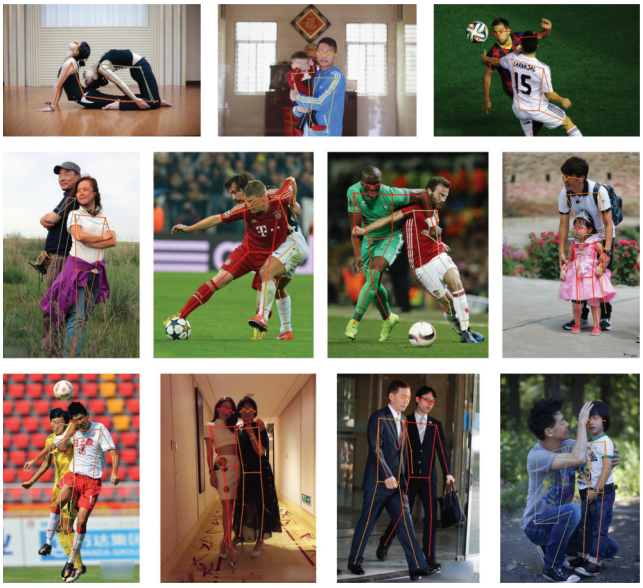

Fig. 9. Qualitative evaluation results for joints detection on overlapped human images from OCHuman [56]. Model: DRPose + HRFormer-b; Input Size: 384×288. Best be viewed in color and zoomed in.

Moreover, DRPose is compared with SOTA methods on the MPII val set with the input size of 256×256, as shown in Table XV. Our DRPose with HRNet-W32 achieves 91.5%/43.6% PCKh@0.5/0.1 score, and outperforms all the SOTA models with less cost of memory. Meanwhile, RPose exceeds the newest model SHaRPose-B by 0.1% PCKh@0.5. Specifically, our method improves the baseline HRNet-W32 by 1.2%/5.9% PCKh@0.5/0.1 score while only needing 0.2M and 0.1 GFLOPs execution costs.

In addition, we compare the proposed DCD with the existing methods of coordinate decoding, as shown in Table XVI. For fair comparison, the backbones of all methods are unified to HRNet-W32, and the input size is set to 256×256 for all models. It can be seen that our method achieves highest

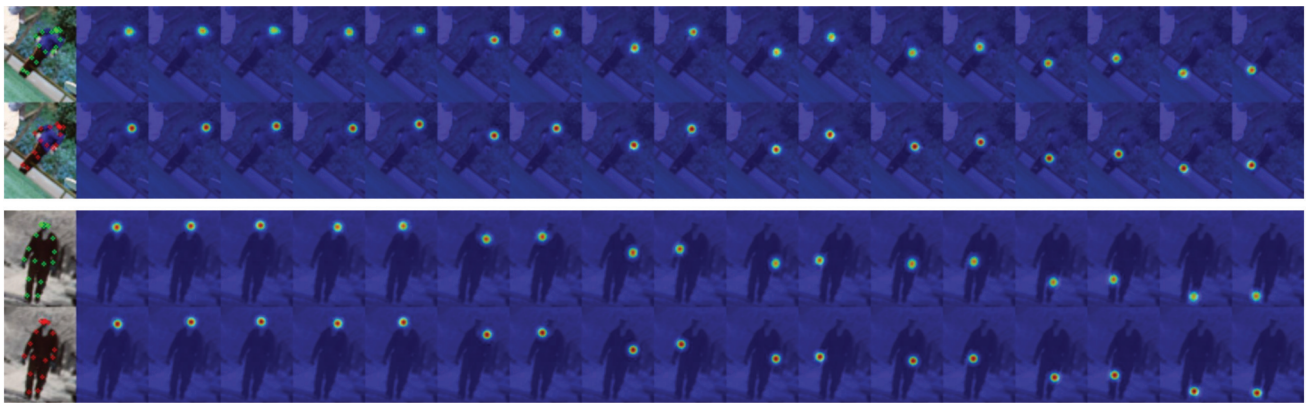

Fig. 10. Visual comparison of the predicted and GT heatmaps and final coordinates inferred on the COCO val set. (Top) Each row is the predicted joints and heatmaps. (Bottom) Each row is their GT counterparts. Method: HRNet-W32+ DRPose; Input size: 256 × 192. Best be viewed in color and zoomed in.

TABLE XVI
COMPARISON WITH OTHER COORDINATE DECODING METHODS ON THE COCO VAL SET

Method	Backbone	Input size	AP	AR
IPR [47]	HRNet-W32	256×192	72.9	-
DARK [11]	HRNet-W32	256×192	75.6	80.8
UDP [46]	HRNet-W32	256×192	75.7	80.8
RB-IPR [48]	HRNet-W32	256×192	75.8	-
PCT [49]	HRNet-W32	256×192	74.7	79.3
ITL [12]	HRNet-W32	256×192	71.3	79.6
DCD (Ours)	HRNet-W32	256×192	**76.2**	**81.3**

prediction accuracy of 76.2%/81.3% in AP/AR, so that the advantage of the proposed DCD is demonstrated.

All the above comparisons with SOTA models have demonstrated the advantage of our proposed DRPose in the HPE task.

D. Qualitative Results

In Fig. 8, we visualize some pose estimated results obtained by performing SimpleBaseline-R50 [28] (red) and DRPose (green) on the COCO val set. From these pictures we can observe that the predicted results of our DRPose are more accurate than the baseline, which further demonstrates the effectiveness of our proposed DRPose. Moreover, to analyze the robustness of the proposed method on more complex scene, we test it on some overlapped human images from OCHuman [56], as shown in Fig. 9. We can observe that our method can detect most joints effectively in the situation of human interaction, but there is still room for improvement.

In Fig. 10, we visualize some results (i.e., heatmaps and final coordinates) predicted by HRNet-W32+ DRPose model and the corresponding GT counterparts. From these results we can observe that the expression of predicted distributions varies, instead of a fixed Gaussian distribution. Therefore, it is valuable to adaptively calculate the final joint location and dynamically adjust the target distribution trained by our DCD.

E. Discussion

Most researchers in HPE strive for high-capacity general networks to enhance the model performance, but neglect the study for some inherent limits in the basic framework. Our experiments focus on confirming the effectiveness of revising the representation and target deviations, as well as validating the usefulness of the proposed modules (i.e., SDB and DCD). Concretely, in Table I, the results show that our work of revising deviations can improve most existing models. Moreover, the applicability of the designed SDB and DCD modules could be verified by Table III. Then, our hypothesis of using the scale factor to adapt scale-dependent features can be demonstrated by Table II. Meanwhile, the meaning of exploiting end-to-end way to alleviate the target deviation is established by Table VII. Furthermore, the ablation studies section includes other experiments that explain the model design in more detail. In addition, the results in Tables XIV–XVI show the progressiveness of our method. Finally, qualitative results (see Figs. 8 and 10) demonstrate that our method performs well in general scene, but struggles to accurately detect human joints in complex scene such as occluded humans (see Fig. 9). Therefore, exploring the robustness of our method in overlapped human images is a significant study direction in the future work.

V. CONCLUSION

In this article, we raise two inherent deviations in the methods of heatmap-based HPE with top-down paradigm, i.e., representation and target deviations, and a novel framework DRPose is proposed to revise them. To mitigate the representation deviation, we present an SDB block to transfer feature maps from multiple scale-dependent domains into a unified intermediate domain. To alleviate the target deviation, a DCD is designed to reform the fixed coordinate decoding to a learnable process end-to-end supervised by the GT coordinate. Extensive experiments on the COCO and MPII benchmarks demonstrate the effectiveness of our proposed method, and our DRPose achieves a new state-of-the-art performance.

Our deviation mitigation framework provides a critical solution to address fundamental limitations in HPE, and we hope our simple and effective design will motivate future efforts in this field.

REFERENCES

[1] J. Liu, N. Akhtar, and A. Mian, "Adversarial attack on skeleton-based human action recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 4, pp. 1609–1622, Apr. 2020.

[2] X. Gao, Y. Yang, Y. Wu, and S. Du, "Learning heterogeneous spatial–temporal context for skeleton-based action recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 12130–12141, Sep. 2024.

[3] Z. Qin et al., "Fusing higher-order features in graph neural networks for skeleton-based action recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4783–4797, Apr. 2024.

[4] M. Dong and C. Xu, "Skeleton-based human motion prediction with privileged supervision," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 10419–10432, Dec. 2023.

[5] J. Fu, F. Yang, Y. Dang, X. Liu, and J. Yin, "Learning constrained dynamic correlations in spatiotemporal graphs for motion prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 10, pp. 14273–14287, Oct. 2024.

[6] J. Yu, Y. Xu, H. Chen, and Z. Ju, "Versatile graph neural networks toward intuitive human activity understanding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 8869–8881, Jul. 2024.

[7] H. Rao et al., "A self-supervised gait encoding approach with locality-awareness for 3D skeleton based person re-identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6649–6666, Oct. 2022.

[8] X. Gu, Y. Guo, F. Deligianni, B. Lo, and G.-Z. Yang, "Cross-subject and cross-modal transfer for generalized abnormal gait pattern recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 546–560, Feb. 2021.

[9] R. Sun et al., "Higher order polynomial transformer for fine-grained freezing of gait detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 12746–12759, Sep. 2024.

[10] J. Li et al., "Human pose regression with residual log-likelihood estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 11025–11034.

[11] F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu, "Distribution-aware coordinate representation for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 7093–7102.

[12] C. Du, Z. Yan, Z. Xiong, and L. Yu, "Boosting integral-based human pose estimation through implicit heatmap learning," *Neural Netw.*, vol. 179, Nov. 2024, Art. no. 106524.

[13] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.

[14] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3686–3693.

[15] Y. Zhang, Y. Bai, M. Ding, S. Xu, and B. Ghanem, "KGSNet: Key-point-guided super-resolution network for pedestrian detection in the wild," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 2251–2265, May 2021.

[16] Y. Zhang, Y. Bai, M. Ding, and B. Ghanem, "Multi-task generative adversarial network for detecting small objects in the wild," *Int. J. Comput. Vis.*, vol. 128, no. 6, pp. 1810–1828, Jun. 2020.

[17] Y. Zhang, Y. Bai, M. Ding, Y. Li, and B. Ghanem, "W2F: A weakly-supervised to fully-supervised framework for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 928–936.

[18] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "Finding tiny faces in the wild with generative adversarial network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 21–30.

[19] H. Ma et al., "PPT: Token-pruned pose transformer for monocular and multi-view human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, Tel Aviv, Israel. Cham, Switzerland: Springer, 2022, pp. 424–442.

[20] D. Wang and S. Zhang, "Contextual instance decoupling for robust multi-person pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11050–11058.

[21] W. Mao et al., "Poseur: Direct human pose regression with transformers," in *Proc. Eur. Conf. Comput. Vis.*, Tel Aviv, Israel. Berlin, Germany: Springer, Oct. 2022, pp. 72–88.

[22] N. Xue, T. Wu, G.-S. Xia, and L. Zhang, "Learning local–global contextual adaptation for multi-person pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13055–13064.

[23] Z. Kan, S. Chen, Z. Li, and Z. He, "Self-constrained inference optimization on structural groups for human pose estimation," in *Proc. 17th Eur. Conf. Comput. Vis. (ECCV)*, Tel Aviv, Israel. Cham, Switzerland: Springer, Jan. 2022, pp. 729–745.

[24] X. Xu, Y. Gao, K. Yan, X. Lin, and Q. Zou, "Location-free human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13127–13136.

[25] H. Wang, L. Zhou, Y. Chen, M. Tang, and J. Wang, "Regularizing vector embedding in bottom-up human pose estimation," in *Proc. 17th Eur. Conf. Comput. Vis. (ECCV)*, Tel Aviv, Israel. Cham, Switzerland: Springer, Jan. 2022, pp. 107–122.

[26] W. McNally, K. Vats, A. Wong, and J. McPhee, "Rethinking keypoint representations: Modeling keypoints and poses as objects for multi-person human pose estimation," in *Proc. 17th Eur. Conf. Comput. Vis. (ECCV)*, Tel Aviv, Israel. Cham, Switzerland: Springer, Jan. 2022, pp. 37–54.

[27] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7103–7112.

[28] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 466–481.

[29] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 5693–5703.

[30] Y. Li et al., "Tokenpose: Learning keypoint tokens for human pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2021, pp. 11313–11322.

[31] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 5998–6008.

[32] Y. Yuan et al., "HRFormer: High-resolution vision transformer for dense predict," in *Proc. Conf. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 7281–7293.

[33] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose: Simple vision transformer baselines for human pose estimation," in *Proc. Conf. Workshop. Neur. Inf. Process. Syst. (NIPS)*, 2022, pp. 38571–38584.

[34] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. ICLR*, Jan. 2021, pp. 1–21.

[35] Y. Dai, X. Wang, L. Gao, J. Song, F. Zheng, and H. T. Shen, "Overcoming data deficiency for multi-person pose estimation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 8, pp. 10857–10868, Aug. 2024.

[36] V. Gupta, A. Yadav, and D. K. Vishwakarma, "HumanPoseNet: An all-transformer architecture for pose estimation with efficient patch expansion and attentional feature refinement," *Expert Syst. Appl.*, vol. 244, Jun. 2024, Art. no. 122894.

[37] X. An, L. Zhao, C. Gong, N. Wang, D. Wang, and J. Yang, "SHaRPose: Sparse high-resolution representation for human pose estimation," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2024, vol. 38, no. 2, pp. 691–699.

[38] L. Wang, Y. Wang, Z. Lin, J. Yang, W. An, and Y. Guo, "Learning a single network for scale-arbitrary super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4801–4810.

[39] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "CondConv: Conditionally parameterized convolutions for efficient inference," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2019, pp. 1307–1318.

[40] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *Proc. Eur. Conf. Comput. Vis.*, Glasgow, U.K. Cham, Switzerland: Springer, Aug. 2020, pp. 282–298.

[41] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8110–8119.

[42] J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, Dec. 2014, pp. 1799–1807.

[43] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2016, pp. 483–499.

[44] G. Papandreou et al., "Towards accurate multi-person pose estimation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 4903–4911.

[45] R. Zhang et al., "Exploiting offset-guided network for pose estimation and tracking," in *Proc. CVPR Workshops*, Jan. 2019, pp. 20–28.

[46] J. Huang, Z. Zhu, F. Guo, and G. Huang, "The devil is in the details: Delving into unbiased data processing for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 5700–5709.

[47] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, "Integral human pose regression," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 529–545.

[48] K. Gu, L. Yang, and A. Yao, "Removing the bias of integral pose regression," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 11047–11056.

[49] Z. Geng, C. Wang, Y. Wei, Z. Liu, H. Li, and H. Hu, "Human pose as compositional tokens," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 660–671.
[50] V. Turchenko, E. Chalmers, and A. Luczak, "A deep convolutional auto-encoder with pooling–unpooling layers in caffe," 2017, *arXiv:1701.04949*.
[51] D. C. Luvizon, H. Tabia, and D. Picard, "Human pose regression by combining indirect part detection and contextual information," *Comput. Graph.*, vol. 85, pp. 15–22, Dec. 2019.
[52] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, Dec. 2015, pp. 2017–2025.
[53] J. Lei Ba, J. Ryan Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
[54] Y. Li et al., "SimCC: A simple coordinate classification perspective for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 89–106.
[55] H. Qu, L. Xu, Y. Cai, L. G. Foo, and J. Liu, "Heatmap distribution matching for human pose estimation," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 24327–24339.
[56] S.-H. Zhang et al., "Pose2Seg: Detection free human instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 889–898.

Zian Zhang received the B.S. and M.S. degrees in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree with the School of Instrumentation Science and Engineering.

He was a Visiting Student with King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia, in 2024. His research interests include computer vision, pattern recognition, machine learning, deep learning, human pose estimation, object detection, and skeleton-based biometric recognition.

Yongqiang Zhang received the M.S. and Ph.D. degrees in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 2015 and 2020, respectively.

He was a Visiting Student with King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia, from 2017 to 2018. He was an Assistant Professor at the School of Instrumentation Science and Engineering, HIT, from 2020 to 2024. Currently, he is a Professor at the College of Computer Science (College of Software-College of Artificial Intelligence), Inner Mongolia University, Hohhot, China. His research interests include computer vision, pattern recognition, machine learning, deep learning, face detection, weakly/fully supervised object detection, activity detection, and image and video understanding in the real world.

Yancheng Bai received the M.S. degree from Sichuan University, Sichuan, China, in 2009, and the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014.

He was a Post-Doctoral Researcher with King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia, from 2016 to 2018. He is an Associate Professor with Pattern Recognition and Intelligent System, Institute of Software, Chinese Academy of Sciences. His research interests include computer vision, pattern recognition, artificial intelligence, machine learning, and deep learning.

Man Zhang received the M.S. degree in optical engineering from Harbin Institute of Technology (HIT), Harbin, China, in 2020, where he is currently pursuing the Ph.D. degree.

His research interests include computer vision, pattern recognition, machine learning, deep learning, object detection, motion forecasting, and 3-D occupancy prediction.

Rui Tian is currently pursuing the M.S. degree with Harbin Institute of Technology (HIT), Harbin, China.

His research interests include computer vision, pattern recognition, machine learning, deep learning, object detection, weakly supervised learning, and contrastive learning.

Yin Zhang received the M.S. degree in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 2021, where he is currently pursuing the Ph.D. degree.

His research interests include computer vision, pattern recognition, machine learning, deep learning, object detection, knowledge distillation, and image super-resolution.

Mingli Ding received the B.S., M.S., and Ph.D. degrees in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 1996, 1997, and 2001, respectively.

He was a Visiting Scholar in France, from 2009 to 2010. Currently, he is a Professor at the School of Instrumentation Science and Engineering, HIT. He has authored or co-authored over 40 papers in peer-reviewed journals and conferences. His research interests include intelligence tests and information processing, automation test technology, computer vision, and machine learning.

Wangmeng Zuo (Senior Member, IEEE) received the Ph.D. degree in computer application technology from Harbin Institute of Technology, Harbin, China, in 2007.

From 2004 to 2006, he was a Research Assistant with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. From 2009 to 2010, he was a Visiting Professor with Microsoft Research Asia, Beijing, China. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology. He has authored or co-authored over 100 papers in top-tier academic journals and conferences. His current research interests include image enhancement and restoration, image/video generation, and image classification.

Dr. Zuo has served as an Associate Editor for IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and a Senior Editor for the *Journal of Electronic Imaging*.