RSPose: Ranking Based Losses for Human Pose Estimation

Muhammed Can Keles[a], Bedrettin Cetinkaya[a], Sinan Kalkan[a,b], Emre Akbas[a,b]

[a]*Department of Computer Engineering, Middle East Technical University, Üniversiteler Mahallesi, Dumlupınar Bulvarı No:1, Ankara, 06800, Turkey*
[b]*ROMER, Middle East Technical University, Üniversiteler Mahallesi, Dumlupınar Bulvarı No:1, Ankara, 06800, Turkey*

Abstract

While heatmap-based human pose estimation methods have shown strong performance, they suffer from three main problems: (P1) Commonly used "Mean Squared Error (MSE) Loss" may not always improve joint localization because it penalizes all pixel deviations equally, without focusing explicitly on sharpening and correctly localizing the peak corresponding to the joint; (P2) heatmaps are spatially and class-wise imbalanced; and, (P3) there is a discrepancy between the evaluation metric (i.e., mAP) and the loss functions. We propose ranking-based losses to address these issues. Both theoretically and empirically, we show that our proposed losses are superior to commonly used heatmap losses (MSE, KL-Divergence). Our losses considerably increase the correlation between confidence scores and localization qualities, which is desirable because higher correlation leads to more accurate instance selection during Non-Maximum Suppression (NMS) and better Average Precision (mAP) performance. We refer to the models trained with our losses as RSPose. We show the effectiveness of RSPose across two different modes: one-dimensional and two-dimensional heatmaps, on three different datasets (COCO, CrowdPose, MPII). To the best of our knowledge, we are the first to propose losses that align with the evaluation metric (mAP) for human pose estimation. RSPose outperforms the previous state of the art on the COCO-val set and achieves an mAP score of 79.9 with ViTPose-H, a vision transformer model for human pose estimation. We also improve SimCC Resnet-50, a coordinate classification-based pose estimation method, by 1.5 AP on the COCO-val set, achieving 73.6 AP.

Keywords: Computer Vision, Deep Learning, Class Imbalance, Human Pose Estimation

1. Introduction

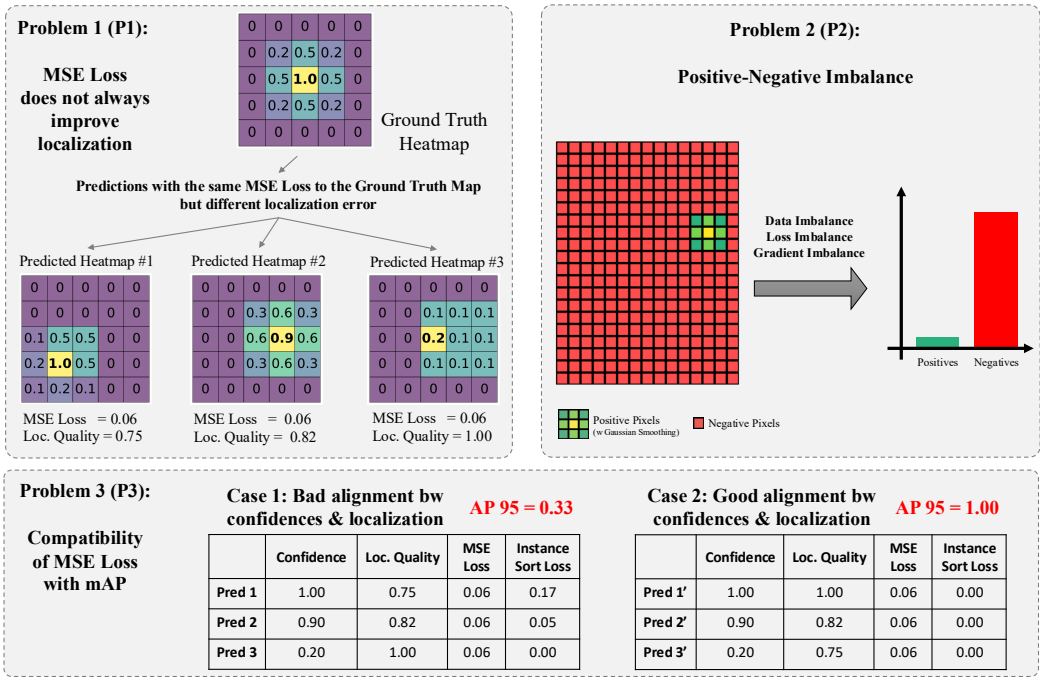
Heatmap-based methods for human pose estimation have been widely adopted due to their strong performance [1, 2, 3, 4]. These methods predict a heatmap for each joint, where each pixel encodes a pseudo-probability indicating the likelihood of the joint being located at that location. Heatmap-based methods have shown better performance compared to regression-based methods. Top-down human pose estimation methods are typically trained by minimizing the Mean-Squared Error (MSE) Loss between the predicted and ground truth heatmaps, where the ground truth heatmap is generated as a 2D Gaussian centered at the annotated joint location. Despite their success, these methods suffer from three key problems (P1-P3):

(P1) MSE Loss may not always improve localization quality. MSE Loss minimizes pixel-wise differences across the entire heatmap, which can lead the model to focus on regions that have little influence on joint localization accuracy (Figure 1).

(P2) Heatmaps incur positive-negative imbalance. Heatmaps used in pose estimation are highly imbalanced, containing a single positive pixel surrounded by a large number of negative ones (as shown in Figure 1 and analyzed in Section 4). For instance, in a common 64×48 heatmap, the positive-to-negative (P/N) ratio is 1:3072. Unless handled, this leads to severe imbalance in loss values as well as the gradients, which considerably affects model performance (analyzed in Section 4). To address this, Gaussian label smoothing combined with MSE loss is commonly employed. While this approach yields good results, it also introduces limitations.

(P3) Discrepancy between the evaluation measure and the training objective. The commonly used evaluation measure, mAP, measures the alignment of instance localization qualities and instance confidences. While MSE supervises the model to learn reasonable confidence scores, it does not directly optimize mAP (Figure 1).

To address these problems, we propose ranking-based loss functions, inspired by their success in object detection and instance segmentation tasks [5, 6, 7, 8, 9, 10, 11], where significant imbalance issues commonly arise. Specifically, we introduce **Spatial-RS Loss**, which trains the model to rank the ground truth logit or pixel above background logits or pixels, which



	Confidence	Loc. Quality	MSE Loss	Instance Sort Loss
Pred 1	1.00	0.75	0.06	0.17
Pred 2	0.90	0.82	0.06	0.05
Pred 3	0.20	1.00	0.06	0.00

	Confidence	Loc. Quality	MSE Loss	Instance Sort Loss
Pred 1'	1.00	1.00	0.06	0.00
Pred 2'	0.90	0.82	0.06	0.00
Pred 3'	0.20	0.75	0.06	0.00

Figure 1: We highlight 3 problems when training heatmap based human pose estimation models with MSE. **(P1)** MSE loss does not always enhance localization, but causes models to learn the non-GT pixels better instead of learning optimal localization. **(P2)** Heatmaps are very sparse and imbalanced. MSE Loss does not provide balanced gradients for positive and negative pixels. **(P3)** MSE Loss does not necessarily optimize the ranking alignment between localization qualities and confidences, which is crucial for the evaluation measure mAP. See Section 5.4 for a detailed discussion on how our methods address these problems.

benefits from the robustness of ranking-based approaches to imbalances in data [6] (**addressing P2**). Furthermore, we propose **Instance-Sort Loss**, which enforces alignment between keypoint confidence scores and their corresponding localization quality (**addressing P1 and P3**). We refer to the models trained with our losses as RSPose.

We validate our approach across both 1D and 2D heatmap-based methods: SimCC-ResNet50 and SimCC-HRNet48 for 1D, and ViTPose-B, ViTPose-H, and HRNet-32 for 2D. Experiments are conducted on three standard benchmarks: COCO, CrowdPose, and MPII. Our method demonstrates strong empirical performance. On the COCO-val set, we achieve 79.9 AP with ViTPose-H, surpassing the previous state of the art. Additionally, our losses significantly increase the correlation between predicted confidence scores and localization quality (16+ percentage points), which is critical for reliable NMS and improved mAP.

Our contributions in this paper are threefold: **(1)** We are the first to successfully adapt and apply ranking based losses for human pose estimation. **(2)** Our ranking-based losses, namely Spatial-RS Loss and Instance-Sort Loss, consistently outperform standard loss functions such as mean-squared-error and KL divergence on a diverse set of models and dataset. **(3)** We introduce Instance Sort Loss (Instance-Sort), a ranking-based objective that encourages stronger alignment between predicted confidence scores and localization accuracy. To our knowledge, this is the first loss in human pose estimation explicitly designed to be aligned with the mAP evaluation measure. Instance-Sort improves confidence-localization correlation and contributes to measurable performance gains.

2. Related Work

Human Pose Estimation. Human pose estimation has shown great progress recently with several different paradigms: top-down [1, 2, 3, 12], bottom-up [13, 14, 15] and end-to-end [16, 17]. Bottom-up methods and end-to-end methods are faster while providing inferior performance. Heatmap based methods have shown superior performance compared to regression based ones, in general.

Commonly, heatmap based methods are trained with MSE Loss that is computed between the ground truth heatmap and the predicted heatmap. Previous work [18] has formulated heatmap prediction as a distribution

matching problem, proposing the usage of Earth Movers Distance as a heatmap loss.

Ranking based losses for computer vision. Ranking based losses have shown strong performance in several computer vision problems such as object detection, instance segmentation, and edge detection. [5, 7, 11, 19]. Ranking based losses deal with both imbalance and uncertainty, as demonstrated in [6, 7, 11].

Comparative Summary. In this work, we propose an alternative loss to MSE for better handling the (i) imbalance, (ii) non-optimal localization, (iii) discrepancy between loss and evaluation metric problems for human pose estimation. To the best of our knowledge, we are the first to propose ranking based losses for human pose estimation and the first to study the loss-to-evaluation metric compatibility for human pose estimation.

3. Pose Estimation: Preliminaries

Pose Estimation Problem Definition.. Given an image $I \in \mathbb{R}^{H \times W \times 3}$, human pose estimation is the problem of estimating N keypoints $\mathcal{K} = \{k_1, ..., k_N\}$ where each keypoint k_i corresponds to a spatial coordinate in I. We use a deep network $f(\cdot; \theta)$ with parameters θ to obtain a heatmap volume $\hat{\mathcal{H}} \in \mathbb{R}^{H \times W \times N}$ which includes one heatmap channel for each keypoint: $\hat{\mathcal{H}} = f(I; \theta)$. We denote the ground truth heatmap with $\mathcal{H} \in \mathbb{R}^{H \times W \times N}$. Although each type of keypoint has a single true-positive pixel, the ground-truth heatmap is typically generated by applying a 2D Gaussian kernel to smooth the label. One dimensional heatmaps are also used. SimCC [12] employs a combination of two independent 1D heatmaps – one for the horizontal (x-axis) and one for the vertical (y-axis) coordinates. To achieve sub-pixel localization precision and mitigate quantization errors inherent in traditional heatmap-based methods, this approach discretizes each axis into $N_x = W \cdot k$ and $N_y = H \cdot k$ bins, where W and H are the image width and height, and k is a splitting factor greater than or equal to 1. The ground truth for each keypoint is represented as a 1D Gaussian distribution centered at the true coordinate, providing a soft label that reflects spatial uncertainty. The model outputs predicted probability distributions $\hat{\mathbf{p}}_x$ and $\hat{\mathbf{p}}_y$ over these discretized bins: $(\hat{\mathbf{p}}_x, \hat{\mathbf{p}}_y) = f(I; \theta)$. We denote the ground truth distributions with $(\mathbf{p}_x, \mathbf{p}_y)$.

Loss Functions for Pose Estimation.. In this paper, we limit our scope to heatmap-based pose estimation methods. The literature has adopted different

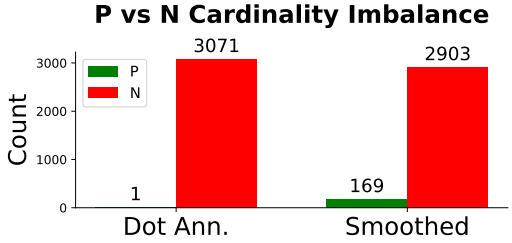


Figure 2: Positive-negative imbalance between positive and negative samples in heatmaps, with an input size of 256×196 and a heatmap size of 64×48. Left: One pixel is positive and other pixels are negative for a joint. Right: A region of pixels around a positive pixel are annotated a degree of positiveness using a Gaussian function centered at the positive pixel.

loss functions for supervising a deep network's heatmap prediction $\hat{\mathcal{H}}$. For example, mean-squared error (MSE) loss is commonly used to penalize pixel-wise predictions as [1, 20]:

$$\mathcal{L}_{\text{MSE}}(\hat{\mathcal{H}}, \mathcal{H}) = \frac{1}{WH} \sum_{x=1}^{W} \sum_{y=1}^{H} \left(\hat{\mathcal{H}}(x,y) - \mathcal{H}(x,y) \right)^2 . \tag{1}$$

Studies using the formulation proposed by SimCC [12] employ the Kullback-Leibler (KL) divergence to quantify the error between the predicted and ground truth distributions:

$$\log \mathbf{p}_x = \log \left(\text{softmax}(\mathbf{z}_x \cdot \beta) \right), \qquad \log \mathbf{p}_y = \log \left(\text{softmax}(\mathbf{z}_y \cdot \beta) \right), \tag{2}$$

$$\mathcal{L}_{\text{KL}} = \frac{1}{N_x} \sum_{i=1}^{N_x} D_{\text{KL}} \left(\log \mathbf{p}_x^{(i)} \,\|\, \mathbf{y}_x^{(i)} \right) + \frac{1}{N_y} \sum_{j=1}^{N_y} D_{\text{KL}} \left(\log \mathbf{p}_y^{(j)} \,\|\, \mathbf{y}_y^{(j)} \right). \tag{3}$$

4. Pose Estimation and the Problem of Imbalance

Imbalance in Cardinalities.. Visual recognition problems often exhibit imbalance between positive ($\mathcal{P}$) and negative ($\mathcal{N}$) samples [21, 22, 23, 24]. Human pose estimation also exhibits imbalance between positive and negative samples: For a joint, there is only a single positive pixel while the rest of the pixels are all negative. This so-called dot annotation is a case of severe imbalance (Figure 2), which can yield sub-optimal performances for the minority class.

This imbalance is often addressed in the literature by applying Gaussian label smoothing at the positive pixels. This way, a region of pixels around a positive pixel is annotated with a degree of positiveness based on their

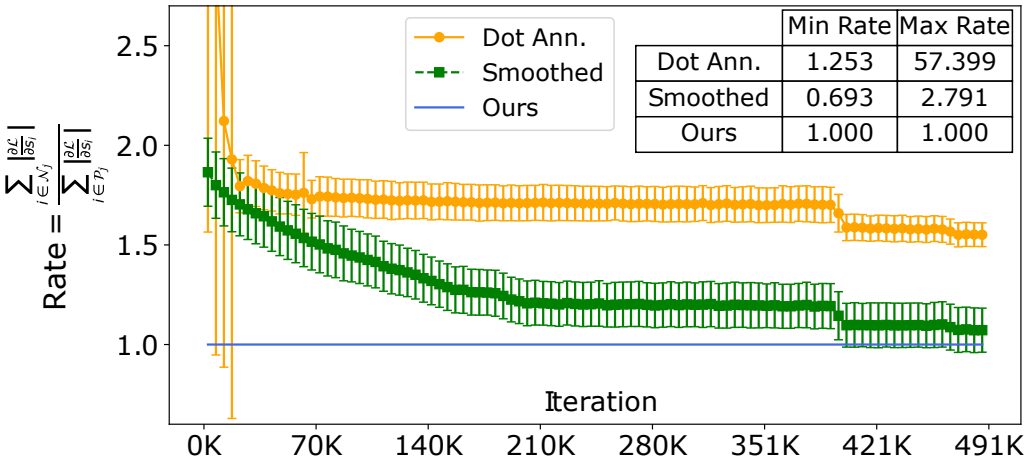


Figure 3: Cardinality imbalance between positive and negative keypoint samples causes imbalance between the gradients of positive and negative keypoint samples. Our proposed method (blue line) addressed this issue, providing a perfect balance between the gradients.

distance to the positive pixel. As illustrated in Figure 2, this produces more positive labels to balance training in favor of positive pixels.

Imbalance in Loss and its Gradients. To better see how the cardinality imbalance between positive ($\mathcal{P}$) and negative ($\mathcal{N}$) samples might affect the training of $f(\cdot;\theta)$, we can rewrite the MSE loss from Eq. 1 as:

$$\mathcal{L}_{\text{MSE}}(\hat{\mathcal{H}}, \mathcal{H}) = \frac{1}{WH}\left(\sum_{(x,y)\in\mathcal{P}} \ell_{xy} + \sum_{(x,y)\in\mathcal{N}} \ell_{xy}\right), \tag{4}$$

where $\ell_{xy} = \left(\hat{\mathcal{H}}(x,y) - \mathcal{H}(x,y)\right)^2$. Since $|\mathcal{P}| \ll |\mathcal{N}|$, the total loss is dominated by background (negative) pixels, even though these pixels contribute little semantic value for keypoint localization.

The imbalance in the loss values, in turn, causes gradient descent to an imbalanced distribution of gradients between positives and negatives, as illustrated in Figure 3.

This imbalance between gradients can hinder optimization, as the model is implicitly incentivized to minimize prediction error over the many non-informative background pixels. To illustrate this, in Table 4, we investigate the effect of using Gaussian label smoothing and show that increasing the number of positive samples provides significant improvements.

Method	Loss	Label Type	Imbalance Ratio ($\lvert\mathcal{N}\rvert/\lvert\mathcal{P}\rvert$)	AP	AR
ViTPose [1]	MSE	Dot Annotation	3071	67.7	77.3
ViTPose [1]	MSE	Gaussian Label S.	17	75.8	81.1
SimCC [12]	KL-Div.	Dot Annotation	along x: 383, along y: 511	71.3	77.3
SimCC [12]	KL-Div.	Gaussian Label S.	along x: 1.2, along y: 2	73.4	80.6

Table 1: The impact of cardinality imbalance on pose estimation performance. Increasing the number of positive samples by using Gaussian label smoothing improves the pose estimation performance. ViTPose uses a ViT-B backbone and SimCC uses a ResNet-50 backbone. (SB: SimpleBaseline [25]).

5. Method

In this section, we describe our novel loss functions for human pose estimation: Spatial Rank & Sort (RS) Loss and Instance Sort Loss. Both of these losses supervise the confidence scores of pixels.

Spatial RS Loss aims to improve the localization quality of keypoints. It has two components: Spatial Rank Loss and Spatial Sort Loss. **Spatial Rank Loss** enforces the confidences of positive pixels to be higher than that of negative pixels. **Spatial Sort Loss** aims to sort positive pixels among themselves respect to their ground-truth labels. That is, a positive pixel with a label, say 0.8, is enforced to have a higher confidence score than another positive pixel with a lower value label.

Instance Sort Loss, on the other hand, aims to adjust confidence scores across instances. For a given joint type, it selects the highest-confidence pixel from each person instance in the batch and sorts them by "keypoint similarity" (KS) score, which measures localization quality (and acts like an intersection over union for keypoints).

5.1. Spatial Rank & Sort Loss

Our first contribution, Spatial Rank & Sort Loss, has two components, Ranking Loss ($\mathcal{L}_{\text{S-Rank}}$) and Sorting Loss ($\mathcal{L}_{\text{S-Sort}}$).

(1) Spatial Rank Loss ($\mathcal{L}_{\text{S-Rank}}$) aims to rank positively labeled pixels above negatively labeled ones (adapting the definitions in Oksuz et al. [7]):

$$\mathcal{L}_{\text{S-Rank}} = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} (\ell_{\text{S-Rank}}(i) - \ell^*_{\text{S-Rank}}(i)), \tag{5}$$

where $\ell_{\text{S-Rank}}(i)$ is the current ranking error for positive pixel i whereas $\ell^*_{\text{S-Rank}}$ is the target ranking error. $\ell_{\text{S-Rank}}(i)$ measures precision for the i^{th} positive

pixel as the ratio of negative pixels having higher confidence than the positive pixel i:

$$\ell_{\text{S-Rank}}(i) = \frac{\#\text{ of false positives with } \hat{\mathbf{p}} > \hat{\mathbf{p}}_i}{\#\text{ of pixels with } \hat{\mathbf{p}} > \hat{\mathbf{p}}_i} = \frac{N_{\text{FP}}(i)}{\text{rank}(i)} = \frac{\sum_{j\in\mathcal{N}} H(x_{ij})}{\sum_{j\in\mathcal{P}\cup\mathcal{N}} H(x_{ij})}, \tag{6}$$

with $x_{ij} = \hat{\mathbf{p}}_j - \hat{\mathbf{p}}_i$ signifying relative ordering between two pixels and $H(\cdot)$ is the step function: $H(x_{ij}) = 1$ if $x_{ij} > 0$ (i.e., $\hat{\mathbf{p}}_j > \hat{\mathbf{p}}_i$) and 0 otherwise (i.e., $\hat{\mathbf{p}}_j \leq \hat{\mathbf{p}}_i$).

The targe ranking is when all positive pixels are ranked above (have higher confidence than) negative pixels, in which case $\ell^*_{\text{S-Rank}} = 0$.

(2) Spatial Sort Loss ($\mathcal{L}_{\text{S-Sort}}$) sorts positively labeled pixels with respect to their labels and hence, aligns (smoothed) labels with confidences. Following RS Loss [7], we define our sorting loss as:

$$\mathcal{L}_{\text{S-Sort}} = \frac{1}{|\mathcal{P}|} \sum_{i\in\mathcal{P}} \left(\ell_{\text{S-Sort}}(i) - \ell^*_{\text{S-Sort}}(i)\right), \tag{7}$$

where $\ell_{\text{S-Sort}}(i)$ and $\ell^*_{\text{S-Sort}}(i)$ are the current sorting error and target sorting error, respectively. We define the sorting error $\ell_{\text{S-Sort}}(i)$ as the average "labels" ($\mathbf{p}$) of pixels with higher "confidence" ($\hat{\mathbf{p}}$) than pixel i (i.e., $H(x_{ij}) = 1$):

$$\ell_{\text{S-Sort}}(i) = \frac{1}{\text{rank}^+(i)} \sum_{j\in\mathcal{P}} H(x_{ij})(1 - \mathbf{p}_j), \tag{8}$$

where $\mathbf{p}_j \in [0, 1]$ is the label of pixel j, and $\text{rank}^+(i) = \sum_{j\in\mathbf{P}} H(x_{ij})$ is the number of positives with higher confidence than pixel i.

The target sorting error $\ell^*_{\text{S-Sort}}(i)$ is defined as the average label $(1 - \mathbf{p}_j)$ over each positive pixel $j \in \mathcal{P}$ with higher confidence ($H(x_{ij}) = 1$) and higher label ($\mathbf{p}_j \geq \mathbf{p}_i$), calculated as:

$$\ell^*_{\text{S-Sort}}(i) = \frac{\sum_{j\in\mathcal{P}} H(x_{ij})[\mathbf{p}_j \geq \mathbf{p}_i](1 - \mathbf{p}_j)}{\sum_{j\in\mathcal{P}} H(x_{ij})[\mathbf{p}_j \geq \mathbf{p}_i]}, \tag{9}$$

where $[P]$, the Iverson Bracket, is 1 if P is True and 0 otherwise.

5.2. Instance Sort Loss

Instance-Sort Loss, $\mathcal{L}_{\text{S-Loss}}$, aims to improve the alignment between keypoint confidences and their localization qualities. For a certain keypoint type, we define it as:

$$\mathcal{L}_{\text{I-Sort}} = \frac{1}{|\mathcal{I}|} \sum_{i\in\mathcal{I}} \left(\ell_{\text{I-Sort}}(i) - \ell^*_{\text{I-Sort}}(i)\right), \tag{10}$$

where $\ell_{\text{I-Sort}}(i)$ and $\ell^*_{\text{I-Sort}}(i)$ are the current instance sorting error and target instance sorting error, respectively. i refers to a specific person instance in the batch $\mathcal{I}$. Instance sorting error relies on the localization quality of the predicted pixel within that heatmap (i.e. the pixel with highest confidence), which is measured by the Keypoint-Similarity score:

$$\text{KS}(i) = \exp\left(\frac{-d^2}{2s^2k^2}\right), \tag{11}$$

where d is the distance of the highest-confidence pixel to the ground truth joint location, s is the instance area, and k is the keypoint fall-off value defined (by the COCO dataset [26]) for this specific keypoint type. We define the instance sorting error $\ell_{\text{I-Sort}}(i)$ as the average "localization quality" of instances with higher "confidence" ($\hat{\mathbf{p}}$) than instance i (i.e., $H(x_{ij}) = 1$):

$$\ell_{\text{I-Sort}}(i) = \frac{1}{\text{rank}(i)} \sum_{j \in \mathcal{I}} H(x_{ij})(1 - \text{KS}(j)). \tag{12}$$

The target instance sorting error is defined as the average Keypoint Similarity score ($1 - \text{KS}(j)$) over each instance $j \in \mathcal{I}$ with higher confidence ($H(x_{ij}) = 1$) and higher KS ($\text{KS}(j) \geq \text{KS}(i)$):

$$\ell^*_{\text{I-Sort}}(i) = \frac{\sum_{j \in \mathcal{I}} H(x_{ij})[\text{KS}(j) \geq \text{KS}(i)](1 - \text{KS}(j))}{\sum_{j \in \mathcal{I}} H(x_{ij})[\text{KS}(j) \geq \text{KS}(i)]}, \tag{13}$$

where $[\cdot]$ is the Iverson Bracket.

Comparison with Spatial Sort Loss. Note that the formulation of the Instance Sort Loss is very similar to the formulation of Spatial Sort Loss. The difference is that, for Instance Sort loss, the targets are keypoint similarity scores calculated between the ground truth coordinates and predicted coordinates. Moreover, Instance Sort Loss takes a single pixel value for every keypoint (or heatmap) while Spatial Sort Loss takes multiple pixels as positives for each keypoint.

5.3. Overall Loss Function and its Optimization

The overall loss function is a weighted sum of the Spatial Rank, Spatial Sort and Instance Sort losses:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{S-Rank}} + \lambda_1 \mathcal{L}_{\text{S-Sort}} + \lambda_2 \mathcal{L}_{\text{I-Sort}}. \tag{14}$$

Optimization. Note that the three introduced loss functions ($\mathcal{L}_{\text{S-Rank}}$, $\mathcal{L}_{\text{S-Sort}}$ and $\mathcal{L}_{\text{I-Sort}}$) include the step function $H(\cdot)$, whose derivative is either zero

or undefined. In practice, $H(\cdot)$ is replaced by an approximation with a defined, non-zero gradient in a certain interval [5]. Furthermore, despite this approximation, the gradient through $H(\cdot)$ is calculated using an error-driven update mechanism [5, 7]. See the supplementary material for more details on optimization of ranking-based loss functions.

5.4. How Our Methods Address the Three Problems (P1-P3)

Here we discuss how our methods address the three problems (**P1-P3**) highlighted in Introduction:

(P1) Addressing loss not measuring localization quality. We address this problem with the help of all three losses ($\mathcal{L}_{\text{S-Rank}}, \mathcal{L}_{\text{S-Sort}}, \mathcal{L}_{\text{I-Sort}}$), which aim to accurately predict positive pixels and their confidences, as well as to align positive pixel confidences with their keypoint similarity scores functioning as a measure of localization quality.

(P2) Addressing positive-negative imbalance. Previous work [6] showed that a ranking-based loss as our $\mathcal{L}_{\text{S-Rank}}$, by definition, provides balance between the gradients of positive and negative pixels, unaffected by the ratio $\mathcal{N}/\mathcal{P}$ or its effects.

(P3) Addressing discrepancy with the evaluation measure. Instance Sort Loss aligns the ordering of estimated confidences (heatmap maximums) and their respective keypoint similarity scores. As pointed out by prior work [27], AP depends on the predicted confidences being rank-consistent with the object keypoint similarity (OKS) scores. Aligning these two quantities would increase AP. We claim that optimizing the alignment between keypoint confidences and their respective localization qualities (Keypoint Similarity) translates to optimizing the alignment between instance confidences and instance localization (OKS) (See the suppl. mat. for a proof).

6. Experiments

Datasets and Setup. We evaluate our method on three widely used benchmarks: COCO [26], CrowdPose [28], and MPII [29]. We apply our ranking-based losses to both 1D classification-based methods (SimCC [12]) and 2D heatmap-based methods (ViTPose [1], HRNet-32 [20]). On COCO, we evaluate both ViTPose and SimCC with various backbones. For CrowdPose and MPII, we use HRNet-32 as a representative 2D baseline.

Performance Metrics. For COCO and CrowdPose, we report mean Average Precision (mAP) and mean Average Recall (mAR). For MPII, we use

Method	Backbone	Loss	mAP	AP^{50}	AP^{75}	AR
Regression-based Methods						
Poseur [30]	HRFormer-B	MLE	78.9	92.0	85.7	-
PCT [2]	Swin-B	CE + L1	77.7	91.2	84.7	-
	Swin-L	CE + L1	78.3	91.4	85.3	-
	Swin-H	CE + L1	79.3	91.5	85.9	-
1D Classification-based Methods						
SimCC [12]	ResNet-50	KL Div.	72.1	89.7	79.8	78.1
		Spatial RS (Ours)	**73.6**	**90.2**	**80.6**	**78.8**
	HRNet-48	KL Div.	75.9	-	-	81.2
		Spatial RS + Instance Sort Loss (Ours)	**76.6**	**90.8**	**83.4**	**81.5**
2D Heatmap-based Methods						
ViTPose [1]	ViT-B	MSE	75.8	90.7	83.2	81.1
		Spatial RS + Instance Sort Loss (Ours)	**76.6**	**90.9**	**83.4**	**81.7**
	ViT-H	MSE	79.1	91.7	85.7	84.1
		Spatial RS + Instance Sort Loss (Ours)	**79.9**	**92.0**	**86.4**	**84.7**

Table 2: Comparison with the state of the art on COCO-val set. Input size: 256x192.

Percentage of Correct Keypoints (PCK). Additionally, we compute Spearman's rank correlation coefficient to evaluate the alignment between predicted confidence scores and localization accuracy.

Implementation Details. We follow the default training protocols of the original models unless otherwise stated. All models are trained for 210 epochs with an input resolution of 256×192. For ViTPose, we use a learning rate of 2e-4 for ViTPose-B and 1e-4 for ViTPose-H, with weight decay of 1e-3. SimCC and HRNet-32 models follow hyperparameter settings from MMPose, except that we tune only the loss-related hyperparameters (delta and loss coefficients). Heatmap resolution is set to 64×48 for 2D methods and 256×2/192×2 for 1D methods.

6.1. Experimental Results

COCO Results. Table 6 summarizes our results on the COCO [26] validation set. We compare our ranking-based losses (Spatial-RS + Instance-Sort) against baseline losses across multiple architectures. Our losses improve

Loss	mAP	AP^{50}	AP^{75}	AP^{E}	AP^{M}	AP^{H}	AR
MSE	67.8	82.4	73.6	77.1	69.0	55.9	76.8
Spatial Rank	67.2	82.4	72.7	76.7	68.5	55.0	76.6
Spatial Rank & Sort	68.6	83.8	74.0	77.8	70.0	56.5	76.7
Spatial RS + Instance-Sort	**68.9**	**84.3**	**74.3**	**78.0**	**70.1**	**57.4**	**76.9**

Table 3: Results on CrowdPose dataset using HRNet [20] with the HRNetV1-W32 backbone.

Loss	PCK	PCK@0.1
MSE	90.0	33.4
Spatial Rank	90.6	31.2
Spatial Rank & Sort	**90.6**	**34.0**

Table 4: Results on MPII dataset using HRNet [20] with the HRNetV1-W32 backbone.

ViTPose-B and ViTPose-H by 0.8 AP each. For SimCC, we observe gains of 1.5 AP with ResNet-50 and 0.7 AP with HRNet-48.

CrowdPose Results. On CrowdPose [28], we train HRNet-32 with our proposed losses. As shown in Table 6.1, our method improves AP by 1.1 over the MSE baseline.

MPII Results. On MPII [29], our method improves HRNet-32 by 0.6 PCK compared to the baseline. Results are summarized in Table 6.1.

Correlation between localization quality and confidence We measure how our losses affect the correlation between localization quality and confidence scores. Previous work [7, 31] has shown that high correlation between localization qualities and confidences scores are beneficial for the evaluation metric mAP and NMS post processing. In Table 6.1 we report correlation coefficients between the localization quality and confidence scores for the ViTPose-H model trained on the COCO dataset. Our results on Table 6.1 show that applying only Spatial RS Loss improves Spearman's

Loss	Spearman Corr.	mAP
MSE	49.9	79.1
Spatial RS	55.5	79.5
Spatial RS + Instance-Sort	**66.5**	**79.9**

Table 5: Instance wise correlation on COCO using ViTPose [1] with the ViTPose-H backbone.

Method	Backbone	Loss Components	mAP	AP^{50}	AP^{75}	AR
1D Classification-based Methods						
SimCC [12]	ResNet-50	KL Div.	72.1	89.7	79.8	78.1
		Spatial Rank	73.0	89.8	80.2	78.4
		Spatial Rank & Sort	**73.6**	90.2	**80.6**	**78.8**
		Spatial RS + Instance-Sort	73.2	**90.4**	80.5	78.6
SimCC [12]	HRNet-48	KL Div.	75.9	-	-	81.2
		Spatial Rank	76.4	90.6	82.9	81.4
		Spatial Rank & Sort	76.5	**90.8**	83.2	81.5
		Spatial RS + Instance-Sort	**76.6**	**90.8**	**83.4**	**81.5**
2D Heatmap-based Methods						
ViTPose [1]	ViT-B	MSE	75.8	90.7	83.2	81.1
		Spatial Rank	75.9	90.7	83.0	81.1
		Spatial Rank & Sort	76.4	90.5	83.1	**81.8**
		Spatial RS + Instance-Sort	**76.6**	**90.9**	**83.4**	81.7
ViTPose [1]	ViT-H	MSE	79.1	91.7	85.7	84.1
		Spatial Rank	79.1	91.6	85.7	84.0
		Spatial RS	79.5	91.4	85.9	84.6
		Spatial RS + Instance-Sort	**79.9**	**92.0**	**86.4**	**84.7**

Table 6: Ablation study on loss components (COCO-val results).

Ranking Correlation considerably. Applying Instance-Sort Loss on top of Spatial RS Loss also improves the results, resulting in an **16.6%** increase in Spearman's Ranking Correlation Coefficient.

6.2. Ablation Study

Impact of Individual Loss Components. Our proposed objective comprises three components: Spatial Rank Loss, Spatial Sort Loss, and Keypoint Similarity Sort (Instance-Sort) Loss. We conduct an ablation study to isolate the contribution of each component to the final model performance. Results are presented in Table 6.1.

Applying Spatial Rank Loss alone leads to consistent improvements in mAP and mAR across most models and backbones. The only exception is observed on CrowdPose, where Spatial Rank Loss underperforms compared to the baseline MSE loss. We attribute this to the high prevalence of human-to-human occlusions in CrowdPose, which may reduce the effectiveness of relative ranking supervision when used in isolation.

Spatial Sort Loss				Instance Sort Loss			
Delta	Coeff.	AP	AR	Delta	Coeff.	AP	AR
	0.5	76.2	81.4		0.25	76.3	81.6
1.0	1.0	76.2	81.4	2.0	0.50	76.5	**81.7**
	2.0	76.2	81.5		1.00	76.4	81.6
	0.5	76.2	81.5		0.25	**76.6**	**81.7**
1.5	1.0	76.1	81.4	3.0	0.50	76.5	**81.7**
	2.0	**76.4**	**81.8**		1.00	76.5	**81.7**

Table 7: Ablation analysis of deltas and coefficients for ViTPose-B on COCO. The Spatial Sort Loss is used together with the Spatial Rank Loss (delta = 0.4, coefficient = 1.0). Additionally, the Instance Sort Loss is combined with the Spatial Rank Loss (delta = 0.4, coefficient = 0.1) and the Spatial Sort Loss (delta = 1.5, coefficient = 2.0).

Adding Spatial Sort Loss further improves performance in all cases, suggesting that explicitly supervising the relative ordering of keypoints reinforces the signal provided by Spatial Rank Loss. Finally, incorporating Instance-Sort Loss yields additional gains in nearly all settings, with the exception of SimCC with ResNet-50, where its contribution is marginally negative. These results underscore the complementary nature of the three loss components.

Effect of Delta and Coefficients. We also investigate the impact of varying the delta (δ) values and weighting coefficients associated with each loss term. Unlike prior work in ranking-based objectives [7, 11, 19], we decouple the deltas used in the rank and sort losses, allowing for more flexible tuning. Table 6.2 reports the performance under different hyperparameter configurations.

6.3. Efficiency Analysis

See the supplementary material for an analysis of the efficiency of the proposed loss functions.

7. Conclusion

In this paper, we propose ranking-based loss functions for human pose estimation to address key limitations of heatmap representations and the commonly used Mean Squared Error (MSE) loss. MSE penalizes all pixel deviations equally, without explicitly emphasizing sharp and accurate localization of joint peaks. Moreover, due to the significant imbalance between positive

and negative pixels, label smoothing is typically required. Another critical issue is the misalignment between the training objective (MSE loss) and the evaluation metric (mAP). Our ranking-based losses mitigate the imbalance and are better aligned with the evaluation metric. We demonstrate the effectiveness of our approach on both one-dimensional and two-dimensional heatmaps, using three different backbones and three benchmark datasets. Our method surpasses the previous state of the art on the COCO-val set, achieving 79.9 AP with ViTPose-H, a vision transformer model. We also improve SimCC ResNet-50, a coordinate classification-based method, by 1.5 AP, reaching 73.6 AP on COCO-val.

Limitations. Although our proposed losses yield better average precision and its per iteration overhead over the traditional MSE looks minimal (1.52 seconds per iteration vs. 1.427 seconds), this overhead might add up when a large model is trained. For example, full training of ViT-H model takes 4.33 days when using our losses, and 4.06 days when using MSE loss. In addition, our losses use $\sim 2\%$ more GPU memory than the MSE loss. Additionally, since our loss function includes three delta and three coefficient parameters, hyperparameter optimization becomes more challenging.

8. Acknowledgments

We gratefully acknowledge the computational resources provided by METU-ROMER, Center for Robotics and Artificial Intelligence, Middle East Technical University, and TUBITAK ULAKBIM High Performance and Grid Computing Center (TRUBA). Emre Akbas and Sinan Kalkan were supported by the Middle East Technical University Scientific Research Projects (BAP) program through project ADEP-312-2024-11485, titled "New Techniques in Visual Recognition."

References

[1] Y. Xu, J. Zhang, Q. ZHANG, D. Tao, ViTPose: Simple vision transformer baselines for human pose estimation, in: Advances in Neural Information Processing Systems, Vol. 35, 2022, pp. 38571–38584.
URL https://proceedings.neurips.cc/paper_files/paper/2022/file/fbb10d319d44f8c3b4720873e4177c65-Paper-Conference.pdf

[2] Z. Geng, C. Wang, Y. Wei, Z. Liu, H. Li, H. Hu, Human pose as compositional tokens, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 660–671.

[3] P. Lu, T. Jiang, Y. Li, X. Li, K. Chen, W. Yang, RTMO: Towards high-performance one-stage real-time multi-person pose estimation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 1491–1500.

[4] H. Liu, Q. Chen, Z. Tan, J.-J. Liu, J. Wang, X. Su, X. Li, K. Yao, J. Han, E. Ding, Y. Zhao, J. Wang, Group pose: A simple baseline for end-to-end multi-person pose estimation, in: International Conference on Computer Vision, 2023, pp. 15029–15038.

[5] K. Chen, W. Lin, J. Li, J. See, J. Wang, J. Zou, AP-loss for accurate one-stage object detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (11) (2020) 3782–3798.

[6] K. Oksuz, B. C. Cam, E. Akbas, S. Kalkan, A ranking-based, balanced loss function unifying classification and localisation in object detection, Advances in Neural Information Processing Systems 33 (2020) 15534–15545.

[7] K. Oksuz, B. C. Cam, E. Akbas, S. Kalkan, Rank & sort loss for object detection and instance segmentation, in: International Conference on Computer Vision, 2021, pp. 3009–3018.

[8] E. Ramzi, N. Audebert, C. Rambour, A. Araujo, X. Bitot, N. Thome, Optimization of rank losses for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence (2025).

[9] Y. Pu, W. Liang, Y. Hao, Y. Yuan, Y. Yang, C. Zhang, H. Hu, G. Huang, Rank-detr for high quality object detection, Advances in Neural Information Processing Systems 36 (2023) 16100–16113.

[10] F. Tang, Q. Ling, Ranking-based siamese visual tracking, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 8741–8750.

[11] B. Cetinkaya, S. Kalkan, E. Akbas, RankED: Addressing imbalance and uncertainty in edge detection using ranking-based losses, in: IEEE/CVF

Conference on Computer Vision and Pattern Recognition, 2024, pp. 3239–3249.

[12] Y. Li, S. Yang, P. Liu, S. Zhang, Y. Wang, Z. Wang, W. Yang, S.-T. Xia, SimCC: A simple coordinate classification perspective for human pose estimation, in: European Conference on Computer Vision, 2022, pp. 89–106.

[13] Z. Geng, K. Sun, B. Xiao, Z. Zhang, J. Wang, Bottom-up human pose estimation via disentangled keypoint regression, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14676–14686.

[14] Z. Luo, Z. Wang, Y. Huang, L. Wang, T. Tan, E. Zhou, Rethinking the heatmap regression for bottom-up human pose estimation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13264–13273.

[15] N. Xue, T. Wu, G.-S. Xia, L. Zhang, Learning local-global contextual adaptation for multi-person pose estimation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 13065–13074.

[16] H. Liu, Q. Chen, Z. Tan, J.-J. Liu, J. Wang, X. Su, X. Li, K. Yao, J. Han, E. Ding, et al., Group pose: A simple baseline for end-to-end multi-person pose estimation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 15029–15038.

[17] J. Yang, A. Zeng, S. Liu, F. Li, R. Zhang, L. Zhang, Explicit box detection unifies end-to-end multi-person pose estimation, in: The Eleventh International Conference on Learning Representations, 2023.

[18] H. Qu, L. Xu, Y. Cai, L. G. Foo, J. Liu, Heatmap distribution matching for human pose estimation, Advances in Neural Information Processing Systems 35 (2022) 24327–24339.

[19] F. Yavuz, B. C. Cam, A. H. Dogan, K. Oksuz, E. Akbas, S. Kalkan, Bucketed ranking-based losses for efficient training of object detectors, in: European Conference on Computer Vision, 2024, pp. 93–109.

[20] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, et al., Deep high-resolution representation learning for visual recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (10) (2020) 3349–3364.

[21] L. Yang, H. Jiang, Q. Song, J. Guo, A survey on long-tailed visual recognition, International Journal of Computer Vision 130 (7) (2022) 1837–1872.

[22] M. Saini, S. Susan, Tackling class imbalance in computer vision: a contemporary review, Artificial Intelligence Review 56 (Suppl 1) (2023) 1279–1335.

[23] K. Oksuz, B. C. Cam, S. Kalkan, E. Akbas, Imbalance problems in object detection: A review, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (10) (2020) 3388–3415.

[24] S. Das, S. S. Mullick, I. Zelinka, On supervised class-imbalanced learning: An updated perspective and some key challenges, IEEE Transactions on Artificial Intelligence 3 (6) (2022) 973–993.

[25] B. Xiao, H. Wu, Y. Wei, Simple baselines for human pose estimation and tracking, in: European Conference on Computer Vision, 2018, pp. 466–481.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: European Conference on Computer Vision, 2014, pp. 740–755.

[27] K. Gu, R. Chen, X. Yu, A. Yao, On the calibration of human pose estimation, in: International Conference on Machine Learning, 2024, pp. 16530–16547.

[28] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, C. Lu, Crowdpose: Efficient crowded scenes pose estimation and a new benchmark, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[29] M. Andriluka, L. Pishchulin, P. Gehler, B. Schiele, 2d human pose estimation: New benchmark and state of the art analysis, in: Proceedings

of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[30] W. Mao, Y. Ge, C. Shen, Z. Tian, X. Wang, Z. Wang, A. v. den Hengel, Poseur: Direct human pose regression with transformers, in: European Conference on Computer Vision, 2022, pp. 72–88.

[31] F. Kahraman, K. Oksuz, S. Kalkan, E. Akbas, Correlation loss: Enforcing correlation between classification and localization, AAAI Conference on Artificial Intelligence (2023).

[32] M. Contributors, Openmmlab pose estimation toolbox and benchmark, `https://github.com/open-mmlab/mmpose` (2020).

Appendix A. Optimization of Ranking Based Losses

The losses $\mathcal{L}_{\text{S-Rank}}$, $\mathcal{L}_{\text{S-Sort}}$ and $\mathcal{L}_{\text{I-Sort}}$ are defined via discrete counts of pairwise relationships between pixel scores, involving the Heaviside step function $H(\mathbf{p}_j - \mathbf{p}_i)$, which is either non-differentiable or has zero (uninformative) derivative. Consequently, the exact derivative

$$\frac{\partial \mathcal{L}_o}{\partial p_i}, \qquad o \in \{\text{S-Rank, S-Sort, I-Sort}\}, \tag{A.1}$$

vanishes on almost all inputs, preventing direct backpropagation.

To address this, we adopt the error-driven update rule of Chen et al. [5], originally developed for AP-Loss, which replaces the true (zero-almost-everywhere) gradient with a sum of pairwise correction terms. Specifically, we approximate

$$\frac{\partial \mathcal{L}_o}{\partial p_i} \;\approx\; \sum_j L^o_{ji}\, t^o_{ji} \;-\; \sum_j L^o_{ij}\, t^o_{ij}, \tag{A.2}$$

where

- L^o_{ij} is called the primary term quantifying *pairwise error contribution* between pixels i and j under loss o. For ranking loss,

$$L^{\text{S-Rank}}_{ij} = \frac{H(p_j - p_i)}{\sum_{k \in P \cup N} H(p_k - p_i)}, \tag{A.3}$$

  and analogously for $\mathcal{L}_{\text{S-Sort}}$ and $\mathcal{L}_{\text{I-Sort}}$.

- $t_{ij}^{o} \in \{0, 1\}$ is a *pair indicator* selecting only the relevant comparisons:

$$t_{ij}^{\text{Rank}} = \begin{cases} 1, & y_i = 1 \wedge y_j = 0, \\ 0, & \text{otherwise}, \end{cases} \quad t_{ij}^{\text{Sort}} = \begin{cases} 1, & y_i = 1 \wedge y_j = 1, \\ 0, & \text{otherwise}. \end{cases} \tag{A.4}$$

Despite the original losses being defined via non-differentiable counts, this approximation yields a usable gradient estimate for end-to-end training. For full derivation, see Chen et al. [5].

Appendix B. Optimizing the alignment between OKS and instance confidence scores

We claim that improving the alignment between keypoint confidences and their localization quality (Keypoint Similarity) leads to better alignment between instance confidence and instance localization (OKS). In Theorem 1, we show that when the covariance between keypoint confidences and their localization quality increases, the covariance between instance confidence and instance localization quality (OKS) also increases. Prior work [27] has shown that mean Average Precision (mAP) depends on the consistency of the ordering between instance confidences and instance localization quality (OKS).

Theorem 1. *Let $L_k \in \mathbb{R}$ denote the localization quality for keypoint k, and let $C_k \in \mathbb{R}$ denote the model's confidence score for keypoint k. Assume independent keypoints, i.e., $Cov(L_{\cdot,j}, C_{\cdot,m}) = 0$ for $j \neq m$. If $\sum_{k=1}^{K} Cov(L_k, C_k)$ increases, then $Cov(OKS, Conf)$ increases, where $OKS = \sum_{k=1}^{K} L_k$ and $Conf = \sum_{k=1}^{K} C_k$.*

Proof. **Proof.** The covariance between OKS and Conf over n instances is given by:

$$\text{Cov}(\text{OKS}, \text{Conf}) = \frac{1}{n} \sum_{i=1}^{n} (\text{OKS}_i - \mathbb{E}[\text{OKS}])(\text{Conf}_i - \mathbb{E}[\text{Conf}]) \tag{B.1}$$

Expanding OKS_i and Conf_i:

$$\text{Cov}(\text{OKS}, \text{Conf}) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{K}\sum_{j=1}^{K} L_{i,j} - \mathbb{E}[\text{OKS}]\right)\left(\frac{1}{K}\sum_{j=1}^{K} C_{i,j} - \mathbb{E}[\text{Conf}]\right) \tag{B.2}$$

Using the linearity of covariance and expectation:

$$\text{Cov}(\text{OKS}, \text{Conf}) = \frac{1}{K^2}\sum_{j=1}^{K}\sum_{m=1}^{K}\frac{1}{n}\sum_{i=1}^{n}\left(L_{i,j} - \mathbb{E}[L_{\cdot,j}]\right)\left(C_{i,m} - \mathbb{E}[C_{\cdot,m}]\right). \tag{B.3}$$

This simplifies to:

$$\text{Cov}(\text{OKS}, \text{Conf}) = \frac{1}{K^2}\sum_{j=1}^{K}\sum_{m=1}^{K}\text{Cov}(L_{\cdot,j}, C_{\cdot,m}) \tag{B.4}$$

Assuming independent keypoints, i.e., $\text{Cov}(L_{\cdot,j}, C_{\cdot,m}) = 0$ for $j \neq m$, we obtain:

$$\text{Cov}(\text{OKS}, \text{Conf}) = \frac{1}{K}\sum_{j=1}^{K}\text{Cov}(L_{\cdot,j}, C_{\cdot,j}) \tag{B.5}$$

Thus, increasing any $\text{Cov}(L_k, C_k)$ increases $\text{Cov}(\text{OKS}, \text{Conf})$. ■ □

Appendix C. Ablation: Effect of Confidence Scaling

Ranking-based losses optimize the relative ordering of predictions, and as a result the raw logits often span a much larger dynamic range than in standard detection models. Since many downstream applications expect confidence scores to lie in a fixed interval for interpretability and stability, we apply a simple per-keypoint min–max normalization using the observed minima and maxima from the COCO `train2017` [26] set. We evaluate the ViT-H [1] model on COCO `val2017` [26] both before and after this scaling. We observe no change in performance—average precision remains at 79.9% and average recall at 84.7%—demonstrating that straightforward range-constrained scaling yields interpretable confidences without sacrificing accuracy.

Appendix D. Efficiency Analysis

Ranking-based losses are often associated with increased computational overhead, particularly due to the large number of pairwise comparisons required. Prior work [19] addressed this by proposing efficient ranking objectives for object detection, enabling their use with large transformer-based backbones. In our setting, the difference in computational cost compared to the baseline is negligible. This is primarily due to the smaller number of negative pairs in pose estimation compared to object detection. To quantify this, we benchmark the per-iteration training time for ViTPose-H [1] using standard MSE Loss versus our full loss (Spatial-RS + Instance-Sort). Experiments are conducted on 2 NVIDIA H100 GPUs with a batch size of 64 per GPU. Averaged over 1,000 iterations, MSE Loss takes 1.427 seconds per iteration, while our proposed loss takes 1.520 seconds. This indicates that our method introduces minimal overhead relative to conventional losses. Compared to MSE in the previously described setting, our method uses only $\sim 2\%$ more GPU memory.

Appendix E. Experimental Details

For the SimCC [12] and HRNet [20] experiments, we follow the official MMPose [32] configurations and use the Adam optimizer. For ViTPose, we also adopt the MMPose [32] configuration, using the AdamW optimizer with a weight decay of 1e-3.

We use a batch size of 64 for all experiments conducted on the COCO and CrowdPose [28] datasets. For experiments on MPII [29], we use a batch size of 16.

A multi-step learning rate schedule is used, with decay steps at epochs 170 and 200, and a decay factor (gamma) of 0.1.

Loss Hyperparameters.. We tune the delta values and weighting coefficients for the *Spatial Rank*, *Spatial Sort*, and *Instance Sort* loss components. The optimal hyperparameter settings used in our experiments are provided in Table Appendix E.

Appendix F. Computational Resources

SimCC with HRNet-48 backbone, HRNet-32, and ViTPose-B were trained on two A100 GPUs; SimCC ResNet-50 used a single A100 GPU. ViTPose-H

Method	Backbone	S. Rank		S. Sort		Ins. Sort	
		Delta	Coeff	Delta	Coeff	Delta	Coeff
ViTPose [1]	ViT-B	0.4	1.0	1.5	2.0	3.0	0.25
ViTPose [1]	ViT-H	0.4	1.0	1.5	2.0	3.0	1.5
SimCC [12]	ResNet-50	0.4	1.0	0.2	0.25	-	-
SimCC [12]	HRNet-W48	0.3	1.0	1.2	0.5	0.5	0.1

Table E.8: Loss hyperparameters (delta and coefficient) for each method and backbone for the COCO dataset.

was trained on two H100 GPUs. Training the largest model, ViTPose-H, on COCO took approximately 4.5 days on two H100 GPUs.