Supplementary Document of Paper "Human Pose Regression with Residual Log-likelihood Estimation"

Appendix

In the supplemental document, we provide:

§A A more detailed explanation of normalizing flows and RealNVP [3].

§B Experiments on MPII dataset.

§C Additional ablation experiments.

§D Visualization of the learn distribution.

§E The derivation of s in RLE.

§F **Pseudocode** for the proposed method.

§G Qualitative results on COCO, MPII and Human3.6M datasets.

§H Extended experiments on retina OCT segmentation dataset.

A. Normalizing Flows

The idea of normalizing flows is to represent a complex distribution $P_\phi(\bar{\mathbf{x}})$ by transforming a much simpler distribution $P(\bar{\mathbf{z}})$ with a learnable function $\bar{\mathbf{x}} = f_\phi(\bar{\mathbf{z}})$. As described in §3.2, the probability of $P_\phi(\mathbf{x})$ is calculated as:

$$\log P_\phi(\bar{\mathbf{x}}) = \log P(\bar{\mathbf{z}}) + \log \left| \det \frac{\partial f_\phi^{-1}}{\partial \bar{\mathbf{x}}} \right|. \quad (1)$$

The function f_ϕ must be invertible since we need to calculate $\bar{\mathbf{z}} = f_\phi^{-1}(\bar{\mathbf{x}})$. In practice, we can compose several simple mappings successively to construct arbitrarily complex functions, *i.e.* $\mathbf{x} = f_\phi(\mathbf{z}) = f_K \circ \cdots \circ f_2 \circ f_1(\mathbf{z})$, where K denotes the number of mapping functions and $\mathbf{z}_K = \mathbf{x}$. The log-probability of $\mathbf{x}$ becomes:

$$\log P_\Theta(\mathbf{x}|\mathcal{I}) = \log P_\Theta(\mathbf{z}|\mathcal{I}) + \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k^{-1}}{\partial \mathbf{z}_k} \right|. \quad (2)$$

RealNVP. In our paper, we adopt RealNVP [3] to learn the underlying residual log-likelihood. RealNVP design each layer f_k as:

$$\begin{aligned} &f_k(\bar{\mathbf{z}}_{k-1,0:d}, \bar{\mathbf{z}}_{k-1,d:D}) \\ =&(\bar{\mathbf{z}}_{k-1,0:d}, \bar{\mathbf{z}}_{k-1,d:D} \odot e^{g_k(\bar{\mathbf{z}}_{k-1,0:d}} + h_k(\bar{\mathbf{z}}_{k-1,0:d})), \end{aligned} \quad (3)$$

	FLOPs	#Params	AP	AP$_{50}$	AP$_{75}$
3×64	1.8M	53.8K	**70.5**	**88.5**	**77.4**
3×128	6.9M	205.8K	70.2	88.5	77.3
3×256	27.3M	804.8K	69.6	87.9	76.5
5×32	**1.3M**	**40.0K**	70.0	88.2	76.8
5×64	5.2M	153.6K	70.3	88.7	77.4

Table 1: **Computation complexity and parameters of RealNVP** during training.

where $g_k, h_k : \mathbb{R}^d \to \mathbb{R}^{D-d}$ are two arbitrary neural networks, D is the dimension of the input vectors, and d is the splitting location of the D-dimensional variable. The $\odot$ operator represents the pointwise product. In order to chain multiple functions f_k, the input is permuted before each step. K is set to 6 in our experiments. In each function f_k, we adopt L_{fc} fully-connected layers with N_n neurons for both g_k and h_k. Each fully-connected layer is followed by a Leaky-RELU [8] layer.

Computation Complexity. The RealNVP model is fast and light-weighted. The computation complexity and model parameters during training are listed in Tab. 1. It is seen that the flow models are computational and storage efficient. The overhead during training is negligible.

B. Experiments on MPII

In multi-person pose estimation, the final mAP is affected by both the location accuracy and the confidence score. To study how RLE affect the location accuracy and eliminate the impact of the confidence score, we evaluate the proposed regression paradigm on MPII [1] dataset. Following previous settings [9], PCK and AUC are used for evaluation. We adopt the same ResNet-50 + FC model for single-person 2D pose estimation. Data augmentations and training settings are similar to the experiments on COCO.

Ablation Study. Tab. 2 shows the comparison among methods using heatmaps, direct regression and RLE. RLE surpasses the direct regression baseline. While MPII is less challenging than COCO, the improvement is still significant on PCKh@0.1 (relative **13.1**%) with high localization accuracy requirement. Compared to the heatmap-based method,

Method	PCKh@0.5	PCKh@0.1	AUC
Direct Regression	83.8	23.6	52.6
SimplePose (Heatmap) [10]	**87.1**	25.4	**56.2**
Regression with RLE	85.5	26.7	55.1
***Regression with RLE**	85.8	**27.1**	55.5

Table 2: **Effect of Residual Log-likelihood Estimation on MPII validation set.**

Method	MPII			Human3.6M	
	PCKh@0.5	PCKh@0.1	AUC	MPJPE	PA-MPJPE
DLE	84.3	25.3	53.5	51.0	39.8
RLE	**85.5**	**26.7**	**55.1**	**48.6**	**38.5**

Table 3: **Comparison between DLE and RLE** on MPII and Human3.6M.

Method	reg. loss weight	hm. loss weight	AP
Direct Regression (ℓ_1)	1	1	57.5
Direct Regression (ℓ_1)	1	0.5	56.7
Direct Regression (ℓ_1)	1	0	58.1
RLE	1	1	70.4
RLE	1	0.5	70.2
RLE	1	0	70.5

Table 4: **Effect of the auxiliary heatmap loss.**

RLE achieves comparable performance (5.1% PCKh@0.1 higher, 1.8% PCKh@0.5 lower and 1.9% AUC lower), and the pre-trained model achieves the best PCKh@0.1 results. RLE shows the superiority in high precision localization.

C. Ablation Study

Comparison between DLE and RLE. In this work, direct likelihood estimation (DLE) refers to the model that only adopts the reparameterization strategy to estimate the likelihood function. The comparison is conducted on COCO [7] validation set in the paper. Here, we provide more comparison results on MPII [1] and Human3.6M [5] datasets (Tab. 3). It is seen that RLE shows consistent improvements over DLE.

Auxiliary Heatmap Loss. In this experiment, we add an auxiliary heatmap loss to the regression model and study its effect. The regression models follow the top-down framework with the "ResNet-50 + FC" architecture. To train the model with the auxiliary loss, the ResNet-50 backbone is followed by 3 deconv layers as SimplePose [10] to generate heatmaps. The deconv layers are parallel to the FC layer. Thus the model can predict both heatmaps and the regressed coordinates. It shows that multi-task loss barely brings performance improvements.

	Ankle	Knee	Hip	Wrist	Elbow	Shoulder	Head
Ankle	135.79	60.2	22.72	86.71	70.05	52.09	50.16
Knee	**91.45**	70.56	23.73	94.64	**72.31**	55.71	53.58
Hip	**87.98**	64.04	28.78	153.02	107	78.98	77.15
Wrist	**80.77**	56.05	27.44	216.17	127.28	74.29	77.85
Elbow	**80.25**	57.46	27.87	212.46	156.66	77.71	68.85
Shoulder	**73.3**	**48.01**	24.5	**146.64**	**113.5**	97.39	**159.67**
Head	**68.71**	44.44	21.62	85.87	69.43	52.25	53.39

Table 5: **Per joint occlusion sensitivity analysis of Integral Pose [9].**

	Ankle	Knee	Hip	Wrist	Elbow	Shoulder	Head
Ankle	**117.86**	**59.69**	**19.76**	**83.93**	**66.68**	**51.08**	**48.83**
Knee	94.96	**68.27**	**20.64**	**92.77**	72.96	**54.24**	**50.69**
Hip	88.39	**57.46**	**19.98**	**139.47**	**100.86**	**75.25**	**75.79**
Wrist	83.06	**53.64**	**21.3**	**200.18**	**125.16**	**73.51**	**74.73**
Elbow	81.45	**55.05**	**24.38**	**208.01**	**154.6**	**76.82**	**67.06**
Shoulder	95.77	54.76	**20.93**	152.28	118.28	**96.01**	162.34
Head	72.27	44.44	**18.65**	**83.34**	**66.14**	**49.81**	**48.73**

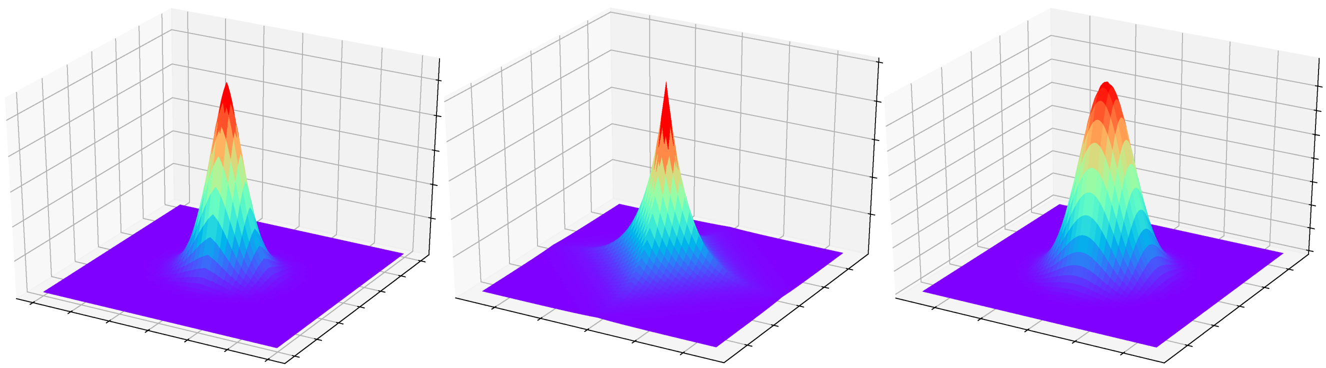
Table 6: **Per joint occlusion sensitivity analysis of RLE.**

Robustness to Occlusion. The regression-based methods predict the body joints in a holistic manner, meaning that they would predict all joints even in cases of occlusions and truncations. In this experiment, we study the impact of occlusion on RLE compared with the heatmap-based method. Similar to PARE [6], we add gray squares on the areas of various joints and study the impact on other joints. Results of Integral Pose [9] and RLE are reported in Table. 5 and Table. 6, respectively. It is seen that RLE improves the occlusion robustness of all joints.

Robustness to Truncation. When facing truncations, regression-based methods can infer the joints outside the input image, while heatmap-based methods failed. This characteristic of regression-based methods makes them robust to crowded cases, where human detection methods are prone to fail. Qualitative comparison between the heatmap-based method and RLE on truncations are shown in Fig. 2. Only the contents inside the bounding boxes are fed to the pose estimation models.

D. Visualization of the Learned Distribution

The visualization of the learned distribution is illustrated in Fig. 1. The learned distribution has a more sharp peak than the Gaussian distribution and a more smooth edge than the Laplace distribution.



(a) The learned distribution (b) Standard Laplace distribution (c) Standard Gaussian distribution

Figure 1: **Visualization** of (a) the learned distribution, (b) Laplace distribution, and (c) Gaussian distribution.

Figure 2: **Qualitative** comparison on truncations. **Top:** RLE. **Bottom:** Heatmap-based SimplePose. Only the contents inside the bounding boxes (blue) are fed to models.

E. Derivation of s in RLE

As Eq. 7 in the paper, we have:

$$\log P_\phi(\bar{\mathbf{x}})d\bar{\mathbf{x}} = \log Q(\bar{\mathbf{x}}) + \log G_\phi(\bar{\mathbf{x}}) + \log s. \tag{4}$$

Thus $P_\phi(\bar{\mathbf{x}}) = Q(\bar{\mathbf{x}})G_\phi(\bar{\mathbf{x}})s$. Since $P_\phi(\bar{\mathbf{x}})$ should be a distribution, its integral equals to one:

$$\begin{aligned}\int P_\phi(\bar{\mathbf{x}}) &= \int Q(\bar{\mathbf{x}})G_\phi(\bar{\mathbf{x}})sd\bar{\mathbf{x}} \\ &= s\int Q(\bar{\mathbf{x}})G_\phi(\bar{\mathbf{x}})d\bar{\mathbf{x}} = 1.\end{aligned} \tag{5}$$

We obtain:

$$s = \frac{1}{\int Q(\bar{\mathbf{x}})G_\phi(\bar{\mathbf{x}})d\bar{\mathbf{x}}}. \tag{6}$$

The integral is approximate by the Riemann sum. Therefore, within the interval $[a, b]$, the value of s can be calculated as:

$$s \approx \frac{1}{\sum_{i=1}^{N} Q(a + i\Delta\mathbf{x})G_\phi(a + i\Delta\mathbf{x})\Delta\mathbf{x}}, \tag{7}$$

Loss	FLOPs of RealNVP	AP	AP_{50}	AP_{75}
DLE	1.8M	62.7	86.1	70.4
RLE ($Q + G$)	1.8M	70.5	88.5	77.4
RLE ($Q + G + s$)	44.2M	70.5	88.6	77.4

Table 7: **Effectiveness of RLE** on COCO validation set. FLOPs in the training phase are reported.

where $\Delta\mathbf{x} = \frac{b-a}{N}$ and N is the total number of subintervals. The interval can set to $[-5, 5]$ in practice, since the value of $Q(\bar{\mathbf{x}})$ is close to zero outside this interval. To accurately calculate s, N should be large enough to obtain a small step $\Delta\mathbf{x}$. In other words, the flow model needs to run N times for calculation, which takes additional computation resources. Interestingly, in our experiments, we find that the term $\log s$ in the loss function is not necessary. As shown in Tab. 7, the effectiveness of RLE over DLE comes from the gradient shortcut in $Q(\bar{\mathbf{x}})$. The term s barely affects the results and can be removed to save computation resources. Therefore, in our implementation, we drop the term $\log s$ for simplicity.

F. Pseudocode for the Proposed Method

The pseudocode of the proposed regression paradigm is given in Alg. 1 (training) and Alg. 2 (inference). It is seen in Alg. 2 that the flow model does not participate in the inference phase. Thus the proposed method won't cause any test-time overhead.

Figure 3: **Qualitative** results on COCO dataset: containing crowded scenes, occlusions, appearance change and motion blur.

Algorithm 1 Pseudocode for training in a PyTorch-like style.

```
# Training
for imgs, gt_mu in train_loader:
   # Regression model predicts 'hat_mu', 'hat_sigma'
       to control the position and scale
   hat_mu, hat_sigma = reg_model(imgs)

   # Calculate the deviation 'bar_mu'
   bar_mu = (gt_mu - hat_mu) / hat_sigma

   # Estimate the log-probability of 'bar_mu' from the
        flow model
   log_phi = flow_model.log_prob(bar_mu)

   if use_residual:
      # Loss for residual log-likelihood estimation
      # Q is the preset density function
      loss = - torch.log(Q(bar_mu)) - log_phi + torch.
           log(hat_sigma)
   else:
      # Loss for direct log-likelihood estimation
      loss = - log_phi + torch.log(hat_sigma)
```

Algorithm 2 Pseudocode for inference in a PyTorch-like style.

```
# Inference
for imgs in test_loader:
   # Run the regression model
   hat_mu, hat_sigma = reg_model(imgs)

   # Calculate the confidence scores
   conf = 1 - torch.mean(hat_sigma, dim=1)

   output = dict(
      coord=hat_mu,
      confidence=conf
   )
```

G. Qualitative Results

Additional qualitative results on COCO, MPII and Human3.6M datasets are shown in Fig. 3, Fig. 4 and Fig. 5.

Method	Mean Error
Direct Regression	18.1
Regression with RLE	**3.1**

Table 8: **Effect of Residual Log-likelihood Estimation on DME dataset.**

H. Experiments on Retina Segmentation

To study the effectiveness and generalization of the proposed regression paradigm, we conduct experiments on boundary regression for retina segmentation from optical coherence tomography (OCT). We evaluate our methods on the publicly available DME dataset [2]. It contains 110 B-scans from 10 patients with severe DME pathology.

We follow the model architecture of the previous method [4] and replace the output layer with a fully-connected layer for regression. The learning rate is set to 1×10^{-4}. We use the Adam solver and train for 200 epochs, with a mini-batch size of 2. Quantitative results are reported in Tab. 8. It shows that RLE significantly reduces the regression error. We hope our method can be extended to more areas and bring a new perspective to the community.

References

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

[2] Stephanie J Chiu, Michael J Allingham, Priyatham S Mettu, Scott W Cousins, Joseph A Izatt, and Sina Farsiu. Kernel regression based segmentation of optical coherence tomogra-

Figure 4: **Qualitative** results on MPII dataset.

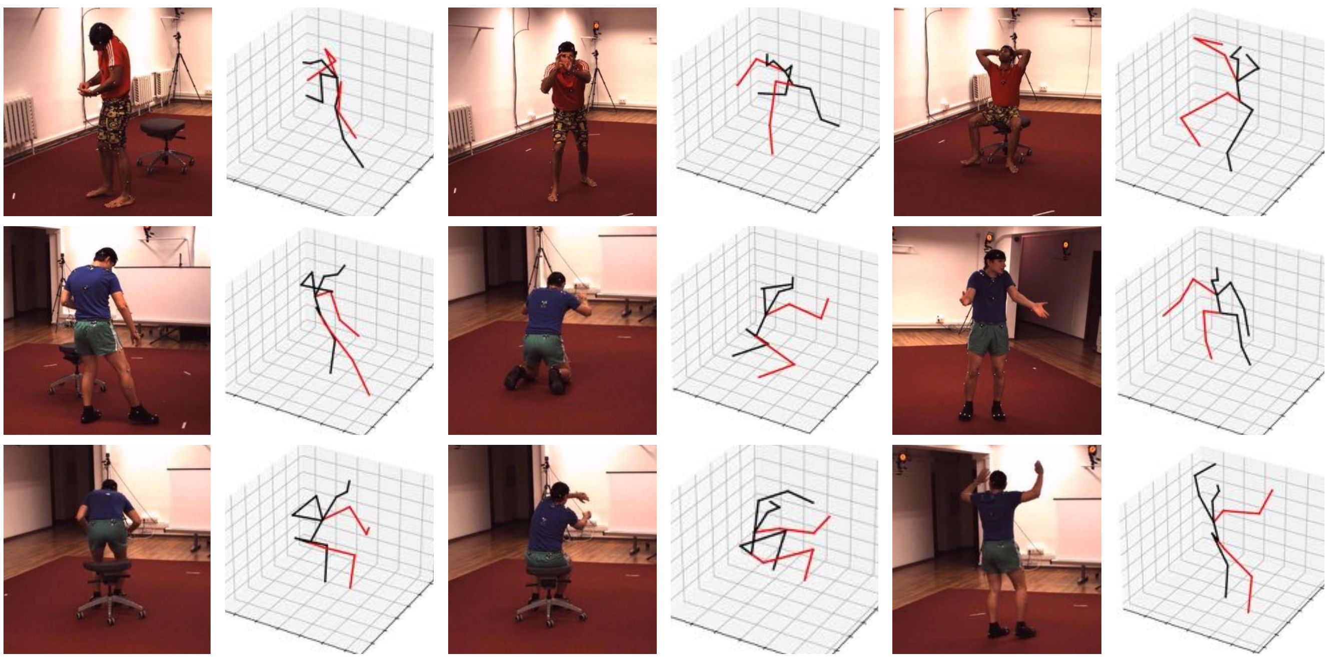


Figure 5: **Qualitative** results on Human3.6M dataset.

phy images with diabetic macular edema. *Biomedical optics express*, 2015. 4

[3] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *ICRL*, 2016. 1

[4] Yufan He, Aaron Carass, Yihao Liu, Bruno M Jedynak, Sharon D Solomon, Shiv Saidha, Peter A Calabresi, and Jerry L Prince. Fully convolutional boundary regression for retina oct segmentation. In *MICCAI*, 2019. 4

[5] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3D human sensing in natural environments. *TPAMI*, 2014. 2

[6] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. Pare: Part attention regressor for 3d human body estimation. *arXiv preprint arXiv:2104.08527*, 2021. 2

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 2

[8] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013. 1

[9] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, 2018. 1, 2

[10] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 2