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

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

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

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

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

$$f_k(z_{k-1,0:d}, z_{k-1,d:D}) = \left( z_{k-1,0:d}, z_{k-1,d:D} \circ e^{g_k(z_{k-1,0:d} + h_k(z_{k-1,0:d}))} \right),$$  (3)

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,
Thus the model can predict both heatmaps and the regressed heatmaps. The deconv layers are parallel to the FC layer. 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.

**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.
The learned distribution
Standard Laplace distribution
Standard Gaussian distribution

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

Ours
SimplePose

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:
\[
\log P_\phi(\bar{x})d\bar{x} = \log Q(\bar{x}) + \log G_\phi(\bar{x}) + \log s. \tag{4}
\]
Thus $P_\phi(\bar{x}) = Q(\bar{x})G_\phi(\bar{x})s$. Since $P_\phi(\bar{x})$ should be a distribution, its integral equals to one:
\[
\int P_\phi(\bar{x}) = \int Q(\bar{x})G_\phi(\bar{x})sd\bar{x} = s \int Q(\bar{x})G_\phi(\bar{x})d\bar{x} = 1. \tag{5}
\]
We obtain:
\[
s = \frac{1}{\int Q(\bar{x})G_\phi(\bar{x})d\bar{x}}. \tag{6}
\]
The integral is approximate by the Riemann sum. Therefore, within the interval $[a, b]$, the value of $s$ can be calculated as:
\[
s \approx \frac{1}{\sum_{i=1}^{N} Q(a + i\Delta x)G_\phi(a + i\Delta x)\Delta x}, \tag{7}
\]
where $\Delta x = \frac{b-a}{N}$ and $N$ is the total number of subintervals. The interval can set to $[-5, 5]$ in practice, since the value of $Q(\bar{x})$ is close to zero outside this interval. To accurately calculate $s$, $N$ should be large enough to obtain a small step $\Delta x$. 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 $Q(\bar{x})$. 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.

<table>
<thead>
<tr>
<th>Loss</th>
<th>FLOPs of RealNVP</th>
<th>AP</th>
<th>AP$_{50}$</th>
<th>AP$_{75}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLE</td>
<td>1.8M</td>
<td>62.7</td>
<td>86.1</td>
<td>70.4</td>
</tr>
<tr>
<td>RLE ($Q + G$)</td>
<td>1.8M</td>
<td>70.5</td>
<td>88.5</td>
<td>77.4</td>
</tr>
<tr>
<td>RLE ($Q + G + s$)</td>
<td>44.2M</td>
<td>70.5</td>
<td>88.6</td>
<td>77.4</td>
</tr>
</tbody>
</table>

Table 7: Effectiveness of RLE on COCO validation set. FLOPs in the training phase are reported.
Algorithm 1 Pseudocode for training in a PyTorch-like style.

```python
# 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.

```python
# 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.

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct Regression</td>
<td>18.1</td>
</tr>
<tr>
<td>Regression with RLE</td>
<td>3.1</td>
</tr>
</tbody>
</table>

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 $1 \times 10^{-4}$. 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


Figure 4: Qualitative results on MPII dataset.

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


