Supplementary: Uncertainty-Aware Adaptation for Self-Supervised 3D Human Pose Estimation

The supplementary document is organized as follows:

• Section 1: Notations
• Section 2: Training algorithms
• Section 3: Network architecture
• Section 4: Qualitative analysis

1. Notations

Most of the notations used in this paper are summarized in Table 1. In the first part, we list the general architecture related notations. Next, we group other notations into a) output of $B_L$, b) output of $B_R$, c) datasets, and finally the adaptation training related notations for both d) pose-level and e) joint-level adaptation.

2. Training algorithms

In this section, we clearly discuss the training algorithms which could not be included in the main paper. Algo. 1 and Algo. 3 show the training algorithm for pose-level and joint-level adaptation respectively. We simultaneously train on samples from all the three datasets, i.e. on $D_s$, $D_t$, and $D_b$ for pose-level adaptation and on $D_s^O$, $D_t^O$, and $D_b^O$ for joint-level adaptation. The pseudo-label selection procedure is clearly explained in both the algorithms (refer Table 1 for a description of the notations). Though we use the above for pose-level adaptation for a fair prior-art benchmarking, one is always free to relax this assumption. a) Under pose-level DA, synthetic training on $D_t^O$ (truncated+full) would make it applicable for both full and truncated target. In Fig. 5, notice the medium level uncertainty elicited by $\text{MRPN}(\text{PU})$ for truncated target (a desirable behaviour). b) On the other hand, joint-level adaptation already suits to both the scenarios ($\text{MRPN}(\text{JU})$) in Fig. 5.

Algo. 2 shows a detailed training procedure to prepare the fusion network for the pose-level adaptation scenario. We prepare a separate fusion network for the joint-level adaptation. Table 3 reports relative contributions of $B_R$ and $B_L$ outputs against the fused. In case of joint level adaptation the loss-term in $L_3$ of Algo. 2 is replaced by $\sum_{j \in J_{\text{out}}} \tilde{w}^{(j)} L_p^{(j)}(\tilde{p}, p_{gt})$ (the second loss-term in $L_6$ of Algo. 3). Similarly, the loss-term in $L_4$ of Algo. 2 is replaced by $\sum_{j \in J_{\text{out}}} \tilde{w}^{(j)} L_{\text{pu}}^{(j)}(\tilde{p}, p_{gt})$ (the second loss-term in $L_{11}$ of Algo. 3).

We trained the framework on an NVIDIA P-100 GPU (16GB) with a batch size of 8. We employ separate Adam

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J$</td>
<td>Total no. of joints (17) indexed by $j$</td>
</tr>
<tr>
<td>$E$</td>
<td>Encoder as the common backbone CNN</td>
</tr>
<tr>
<td>$B_L$</td>
<td>Localization branch (outputs heatmaps)</td>
</tr>
<tr>
<td>$B_R$</td>
<td>Regression branch (outputs 3D pose)</td>
</tr>
<tr>
<td>$T_{FK}$</td>
<td>Forward-kinematics operation</td>
</tr>
<tr>
<td>$T_w$</td>
<td>Weak-perspective projection operation</td>
</tr>
</tbody>
</table>

### Table 1. Notation Table

### Dataset

- $D_s$: Labeled source and unlabeled target datasets (full-body)
- $D_t$: Source and target datasets with occlusion/truncation
- $D_b$: A dataset of background images (other than human)

### Pose-uncertainty

- $u(I)$: Pose-level uncertainty for a given image
- $u_s(I)$: Supervised loss on $D_s$ samples (minimized)
- $u_t(I)$: Pose-uncertainty of $D_t$ samples (maximized)
- $u_b(I)$: Pose-uncertainty of $D_b$ samples (minimized)

### Occlusion-aware supervised loss

- $L_{\text{sup}}^{(j)}$: Loss on pseudo-label target subset $D_{pl}^j$ (minimized)
- $\alpha_{pl}^j$: Threshold to select pseudo-labeled target subset $D_{pl}^j$

### Joint uncertainty

- $u(I,j)$: Joint-level uncertainty (JU) for a given image, joint-id pair
- $u_t(I,j)$: Occlusion-aware supervised loss on $D_{pl}^j$ (minimized)
- $\mathcal{H}_{\text{out}}$: Joint uncertainty in $D_{\text{out}}^j$ (maximized)

### Thresholds

- $\alpha_{\text{out}}^j$: Threshold to select pseudo-labeled target out-view set $J_{\text{out}}^j$


Algorithm 1 Training algorithm for pose-level adaptation.

1: **Input:** Labeled source dataset \(D_s\), unlabeled target dataset \(D_t\), and the background dataset \(D_b\). Let \(\Theta\) denote the learnable parameters of the MRP-Net architecture (excluding the fusion network).
2: **while** \(\text{iter} < \text{MaxIter} \) **do**
   
   **A. Pseudo-label update (after each \(K_{interval}\)).**
   
   3: if \(\text{iter} \mod K_{interval} = 0\) then
   
   4: **Compute** \(D_t^p\) where \(\hat{q}_t^p\) and \(\hat{q}_t^p\) are obtained using current state of network parameters \(\Theta\), as follows:
   
   \[D_t^p = \{I_t : (\hat{q}_t - F_q(\hat{q}_t^p)) + (\hat{q}_t - F_q(\hat{q}_t^p)) < \alpha^p\}\]
   
   5: **end if**

   **B. Adaptation training (for pose-level adaptation).**
   
   6: **Update** \(\Theta\) by minimizing \(L_h(h, h_{gt}), L_p(p, p_{gt})\), and \(U(t)\) (i.e. the first two terms under \(L_{Sup}^{(s)}\)) on a mini-batch of \(D_t\) using separate Adam optimizers.
   
   7: **Update** \(\Theta\) by maximizing \(U(t)\) on a mini-batch of \(D_b\) using Adam optimizer.
   
   8: **Update** \(\Theta\) by minimizing \(U(t)\) on a mini-batch of \(D_t\) using Adam optimizer.
   
   9: **Update** \(\Theta\) by maximizing \(\sum_{j} \tilde{w}(j)L_h(h_{gt})\) and \(\sum_{j} \tilde{w}(j)\) (i.e. the two terms under \(L_{PSup}^{(t)}\)) using separate Adam optimizers.

10: **end while**

Algorithm 2 Training algorithm for the fusion network.

1: **Input:** Labeled source dataset \(D_s\) and the pseudo-labeled target subset \(D_t^p\). The network takes 3 inputs: a) 3D pose predictions via \(B_R\) (i.e. \(\hat{p}\)), b) 2D pose prediction via \(B_L\) (i.e. \(\hat{q}\)), and c) the joint-confidence \(\tilde{w}\) via \(B_L\). Let \(\Theta^f\) denote the learnable parameters of the fusion network.
2: **while** \(\text{iter} < \text{MaxIter} \) **do**
   
   3: **Update** \(\Theta^f\) to minimize \(L_p(p, p_{gt})\) on a mini-batch of \(D_s\) using Adam optimizer.
   
   4: **Update** \(\Theta^f\) to minimize \(\sum_{j} \tilde{w}(j)L_j(\hat{p}_j, p_{gt}^j)\) on a mini-batch of \(D_t^p\) using Adam optimizer.

5: **end while**

Algorithm 3 Training algorithm for joint-level adaptation.

1: **Input:** Labeled source dataset \(D_s^O\), unlabeled target dataset \(D_t^O\), and the background dataset \(D_b\). Let \(\Theta\) denote the learnable parameters of the MRP-Net architecture (excluding the fusion network).
2: **while** \(\text{iter} < \text{MaxIter} \) **do**
   
   **A. Pseudo-label update (after each \(K_{interval}\)).**
   
   3: if \(\text{iter} \mod K_{interval} = 0\) then
   
   4: **Compute** \(J_{inv}^t\) and \(J_{outV}^t\), where \(\hat{q}_t^p\) and \(\hat{q}_t^p\) are obtained using the current state of the network parameters \(\Theta\), as follows:
   
   \[J_{inv}^t = \{(I_t, j) : H(I_t, j) \in D_t^p\} < \alpha^p\]
   
   \[J_{outV}^t = \{(I_t, j) : H(I_t, j) \in D_t^p\} > \alpha^p\]
   
   5: **end if**

   **B. Adaptation training (for joint-level adaptation).**
   
   6: **Update** \(\Theta\) by minimizing \(I_{(t, j) \in J_{inv}^t} L_h(h, h_{gt})\) and \(I_{(t, j) \in J_{outV}^t} L_p(p, p_{gt})\) (i.e. the first two terms under \(L_{Sup}^{OA}\)) on a mini-batch of \(D_s^O\) using separate Adam optimizers.
   
   7: **Update** \(\Theta\) to maximize \(H_j^{(s)} I_{J_{inv}^t} = I_{(t, j) \in J_{inv}^t} H(I, j)\) on a mini-batch of \(D_s^O\) using Adam optimizer.
   
   8: **Update** \(\Theta\) to maximize \(H_j^{(b)} I_{J_{outV}^t} = I_{(t, j) \in J_{outV}^t} H(I, j)\) on a mini-batch of \(D_s^O\) using Adam optimizer.
   
   9: **Update** \(\Theta\) to minimize \(H_j^{(t)} I_{J_{inv}^t} = I_{(t, j) \in J_{inv}^t} H(I, j)\) on a mini-batch of \(D_t^O\) using Adam optimizer.
   
   10: **Update** \(\Theta\) to maximize \(H_j^{(t)} I_{J_{outV}^t} = I_{(t, j) \in J_{outV}^t} H(I, j)\) on a mini-batch of \(D_t^O\) using Adam optimizer.
   
   11: **Update** \(\Theta\) to maximize \(\sum_{j \in J_{inv}^t} \tilde{w}(j) L_j(\hat{h}, h_{gt})\) and \(\sum_{j \in J_{outV}^t} \tilde{w}(j) L_j(\hat{h}, h_{gt})\) (i.e. the two terms under \(L_{PSup}^{OA}\)) using separate Adam optimizers. Here, \(\tilde{w}(j)\) is normalized such that \(\sum_{j \in J_{inv}^t} \tilde{w}(j) = 1\).

12: **end while**

Importance of OOD images. We would like to reiterate that the background images represent an objective segregation of hard-OOD samples. The poses outside of the training distribution are critical to identify and we segregate them via the pseudo-label subset selection criteria (Eq. 4). Eq. 5 selectively imposes a strong loss on the more confident target samples. It is to be noted that, such segregation is highly subjective, and treating these soft-OOD samples as hard-OOD deteriorates the generalization performance.

3. Network architecture

The architecture consists of an ImageNet initialized ResNet-50 (till Res-4F) which bifurcates into two branches, \(B_L\) and \(B_R\) as shown in Fig. 3. \(B_L\) is a convolutional decoder consisting of an alternate series of transposed convolution and general convolution which progressively increases the spatial resolution from \(7 \times 7\) to \(56 \times 56\). The final output of \(B_L\) is 17 heatmap PDFs, \(\hat{h}\) obtained via spatial softmax. These are then used to extract the correspond-
<table>
<thead>
<tr>
<th>Asset used</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human3.6M [1]</td>
<td>Limited license for academic use</td>
</tr>
<tr>
<td>MPI-INF-3DHP [3]</td>
<td>Limited license for academic use</td>
</tr>
<tr>
<td>3DPW [7]</td>
<td>Limited license for academic use</td>
</tr>
<tr>
<td>HumanEva [5]</td>
<td>Limited license for academic use</td>
</tr>
<tr>
<td>SURREAL [6]</td>
<td>Limited license for academic use</td>
</tr>
</tbody>
</table>

Table 3. Relative contribution of fusion network inputs on 3DPW, MPJPE (↓).

<table>
<thead>
<tr>
<th>Methods</th>
<th>η</th>
<th>η + w</th>
<th>Fused</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours(3DPW)</td>
<td>111</td>
<td>122</td>
<td>115</td>
</tr>
<tr>
<td>Ours(JU: H → 3DPW)</td>
<td>116</td>
<td>142</td>
<td>135</td>
</tr>
</tbody>
</table>

Figure 1. Hyperparameter sensitivity.

Figure 2. Qualitative analysis. 3D poses shown correspond to the original camera view and another azimuthal view at +30° or -30° depending on best viewing angle. For results in panel E and F the joints with uncertainty greater than a prefix threshold are highlighted with red-blobs. The model fails on rare poses, complex inter-limb occlusion and heavy background clutter as highlighted by red bases.

Figure 3. Detailed architecture of the proposed MRP-Net. On the right we show the legend. Here, K3C256S2 denotes specifications of the convolutional layer, i.e. 3×3 filter size, 256 filters applied with a stride 2. Here, TConv denotes transposed convolution operation. FC denotes fully-connected layer. x2 and x3 depict number of residual blocks that are stacked to form the corresponding branch.

Fig. 3 shows the detailed architecture. Further, ablation performance with fusion network is shown in Table 4 (MPJPE of #5-7, Table 4). We see that a better adaptation further enhances the gain from fusion network.
Figure 4. A. Shows histogram of the predicted joint-uncertainties for the true in-view and out-view joints separately for source (i.e. inV-S and outV-S) and target (i.e. inV-T and outV-T). BG denotes the histogram of all out-view joints for backgrounds. The shaded regions in the bottom panel depicts $J_{\text{inV}}$ and $J_{\text{outV}}$ which are segregated using the preset thresholds $\alpha_{\text{th}}^0$ and $\alpha_{\text{th}}^h$ respectively (edges of the green-box). Our adaptation algorithm succeeds to separate inV-T and outV-T over the course of adaptation training. B. Shows a similar analysis for pose-uncertainties. We show 5 different examples sampled from different regions of the histogram-bins. Results on right-panel: Notice that to maximize pose-uncertainty for backgrounds (OOD samples), MRPN estimates the 2D landmarks and 3D pose points separated towards opposite diagonal corners. Here, the 2D landmarks are collapsed to the top-left corner whereas the root joint (pelvis) of the model-based 3D predictions are seemed to have collapsed towards the bottom-right corner. Result on bottom-panel: For uncertain target instances, we see two peaks in the joint heatmap PDFs; one at the top-left corner (OOD-related) and the other near the actual joint location. During adaptation, the OOD-related peak suppress while the joint-related peak rises to simultaneously reduce the uncertainty while converging towards the true pose outcome. Results on the left panel: Joint-level uncertainty is indicated by the entropy of heatmap PDF.

Figure 5. Every pose prediction of MRPN is associated with a measure of uncertainty barometer. The barometer height indicates high uncertainty. The blue, green and orange barometers indicate the average prediction uncertainty for the full-pose, true-in-view joints and true-out-view joints respectively. The dotted gray rectangles highlight the failure cases of LCR++ in predicting the correct 3D inter-limb depth though the 2D landmarks align with the GT. In the last 2 rows, the filled red-box under GT column segregates the true out-view joints. The in-view joint predictions of MRPN(JU) (unfilled green rectangles) performs better against the same for LCR++ (unfilled red rectangles) when compared against the same under GT.

4. Qualitative analysis

We perform a thorough qualitative study to interpret the behaviour of our network for a wide variety of in-distribution and out-of-distribution samples (see Fig. 2).

<table>
<thead>
<tr>
<th>No.</th>
<th>Method</th>
<th>$L_{\text{sup}}^{(\epsilon)} - L_{\text{sup}}^{(b)}$</th>
<th>$\gamma$</th>
<th>w/o fuse</th>
<th>w/ fuse</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.</td>
<td>B2(S→H)+DANN</td>
<td>only $L_{\text{sup}}^{(\epsilon)}$</td>
<td>Standard DA</td>
<td>116.8</td>
<td>114.5 (2.3↓)</td>
</tr>
<tr>
<td>6.</td>
<td>B2(S→H)</td>
<td>✓</td>
<td>-</td>
<td></td>
<td>122.4</td>
</tr>
<tr>
<td>7.</td>
<td>B2(S→H)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>113.4</td>
</tr>
</tbody>
</table>

The analysis in Fig. 4A shows that the proposed joint-level adaptation algorithm succeeds to separate inV-T and outV-T over the course of adaptation training, thereby aligning these with inV-S and outV-S respectively. In Fig. 5, MRPN(B1) indicates the occlusion-aware network before the adaptation training. MRPN(PU) and MRPN(JU) indicate the final networks after the pose-level and joint-level adaptations. Further we show the ground-truth (2D) and predictions on LCR++ [4]. MRPN(PU) is not tuned to work on occluded/truncated images and thus yields a higher uncertainty for the last two rows. Whereas, the uncertainty predictions of MRPN(JU) for the green and orange barometer yield the expected behaviour.

References


