Deep Reinforcement Learning for Active Human Pose Estimation
Supplementary Material

Erik Gärnter†, Aleksis Pirinen†, Cristian Sminchisescu†, 2
†Department of Mathematics, Faculty of Engineering, Lund University
2Google Research
{erik.gartner, aleksis.pirinen, cristian.sminchisescu}@math.lth.se

In this supplemental we provide additional insights into our Pose-DRL model. Details of the network architecture are provided in § 1. Further model insights and dataset details are provided in § 2. A description of how we handle missed detections or failed matchings are given in § 3. Finally, additional visualizations are shown in § 4.

1 Model Architecture
See Fig. 1 for a description of the Pose-DRL architecture. The underlying pose estimation networks, DMHS (Popa, Zanfir, and Sminchisescu 2017) and MubyNet (Zanfir et al. 2018), as well as our agent were implemented in Caffe (Jia et al. 2014) and MATLAB. For the Faster R-CNN detector (Ren et al. 2015) we used a publicly available Tensorflow (Abadi et al. 2016) implementation,† with ResNet-101 (He et al. 2016) as base feature extractor.

10% best 10% worst Rest

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Val</th>
<th>Test</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mafia</td>
<td>53,100</td>
<td>27,900</td>
<td>33,728</td>
<td>114,728</td>
</tr>
<tr>
<td>Ultimatum</td>
<td>27,960</td>
<td>4,340</td>
<td>55,825</td>
<td>88,125</td>
</tr>
<tr>
<td>Pose</td>
<td>51,079</td>
<td>29,672</td>
<td>59,288</td>
<td>140,039</td>
</tr>
<tr>
<td>All</td>
<td>132,139</td>
<td>61,912</td>
<td>148,841</td>
<td>342,892</td>
</tr>
</tbody>
</table>

Table 1: Pose-DRL agent’s selection statistics of good / bad viewpoints on the test set splits. The agent consistently chooses a high percentage of good cameras while avoiding bad cameras. Note that randomly choosing cameras would result in always having 10% chosen among the 10% best cameras, and similar for the 10% worst cameras.

2 Additional Insights and Details

More about runtimes. All experiments reported in this supplementary material and in the main paper were performed using an Ubuntu workstation using a single Titan V100. Training the Pose-DRL policy from scratch took about 70 hours after having pre-computed all DMHS / MubyNet features, Faster R-CNN bounding boxes and instance features. When presenting the runtimes (see Figure 2 in the main paper) we include the time needed to compute these detections and features.

Quality of selected viewpoints. To obtain further insights into which cameras the agent is selecting on average, we tracked how often the agent selects good vs bad viewpoints (for the DMHS-based model). Specifically, for each selected camera in the various test set splits, we sorted it into being in either the 10% best or worst cameras based on associated individual reconstruction error. The results are shown in Table 1. It can be seen that the agent typically selects among the best while avoiding the worst viewpoints. The viewpoint errors are more uniform for the single-people Pose scenes, since there are no viewpoints where the target is occluded, hence the camera selection statistics are also more uniform for Pose.

Further dataset insights. In Table 2 we show how we randomly split the Panoptic dataset (Joo et al. 2015) into train, test, and validation sets.
3 Handling Missed Detections or Matchings

For an overview of how we detect and match multiple people, refer to §3.2 in the main paper. In this section we describe what happens in case some persons are not detected or matched. For the detection-based DMHS-version of Pose-DRL, if in a viewpoint there are no detections, or if no detection has a matching cost below the threshold $C$, the underlying pose estimator is computed on the entire input image to obtain a base state descriptor $B_t$ for decision making (no associated pose is fused in this case).

It is possible that one or several persons are not detected in a single viewpoint in an active-view. In this case the pose estimate is set to the fused estimate from the previous active-view as a backup. In case a previous estimate also does not exist (could happen e.g. in the initial active-view of an active-sequence), to be able to compute a reconstruction error we set a placeholder pose estimate where each joint is equal to the center hip location of the ground-truth. Naturally, this is an extremely poor and implausible estimate, but rally by adding the previous fused estimate; see eq. (1) in the main paper.

\[ \bar{x} = f(x_1, \ldots, x_t) \]  

4 Additional Visualizations of Pose-DRL

In Fig. 2 - 3 we show two additional visualizations of how Pose-DRL performs single-target pose estimation in active-views from the Panoptic (Joo et al. 2015) test set we have used in this work. We use SMPL (Loper et al. 2015) for the 3d shape models (both here and in the main paper), and use per-joint median averaging for fusing poses. As it is referenced in the visualizations, we show the equation for a partially fused pose (for the first $j$ steps) within an active-view:\footnote{For active-sequence processing, the agent also fuses temporally by adding the previous fused estimate; see eq. (1) in the main paper} below:

4.1 Using Pose-DRL the Wild

The dense CMU Panoptic studio provides a powerful environment for training and evaluating our proposed model, however it is also interesting to test the model’s applicability in the real world. To this end we captured data with an off-the-shelf smartphone and used internal sensors to estimate the camera pose matrix for each image. This simple process of walking around subjects while they stand still emulates the time-freeze setup in Panoptic and allows us to test our model in the real world. Please note that neither the 3d pose estimation network nor the policy was re-trained; only the instance detector was refined to produce accurate appearance models for the detected people. See Fig. 4 for resulting visualizations. Please note we obtained consent from the people shown.
Figure 2: Visualization of how Pose-DRL performs single-target reconstruction on an active-view (set of viewpoints for a time-freeze) for a Mafia test scene. In this case the agent sees three viewpoints prior to automatically continuing to the next active-view. The reconstruction error reduces from 168 to 107 mm/joint. Left: Viewpoints seen by the agent, where blue marks the current viewpoint (camera) and red marks previous viewpoints. Note that the initial camera was given randomly. Middle: Input images associated to the viewpoints, also showing the detection bounding box of the target person in red – detections for the other people are left out to avoid visual clutter. Right: SMPL visualizations of the partially fused poses, cf. (1). The target person is only partially visible in the initial viewpoint, and the associated pose estimate is inaccurate with the reconstruction incorrectly tilting forward. As the agent visits more viewpoints, the stance of the reconstruction becomes straighter and more correct. The person is fully visible in the final viewpoint, and the associated final fused estimate is accurate and plausible.
Figure 3: Visualization of how Pose-DRL performs single-target reconstruction on an active-view (set of viewpoints for a time-freeze) for an Ultimatum test scene, where in this case the detection and matching is incorrect for the second viewpoint. Left: Viewpoints seen by the agent, where blue marks the current viewpoint (camera) and red marks previous viewpoints. Note that the initial camera was given randomly. Middle: Input images associated to the viewpoints, also showing the detection bounding box of the target person in red – detections for the other people are left out to avoid visual clutter. In the second viewpoint with the incorrect detection and matching, the target person is indicated with a dashed red bounding box, and the incorrect detection used in the pose fusion is shown in yellow. Right: SMPL visualizations of the partially fused poses, cf. (1). The target person is viewed from a suboptimal direction in the first viewpoint, causing the associated pose estimate to be incorrectly tilted. As the agent moves to the next viewpoint to get a better view of the person, the underlying detection and matching system suggests an incorrect detection to feed the pose estimator, which causes the fused estimate to deteriorate severely. However, the agent is able to remedy this by selecting two more good and diverse viewpoints where the target is clearly visible, yielding a considerably better fused pose estimate. In this example the agent sees four viewpoints prior to automatically continuing to the next active-view. The reconstruction error reduces from 149 to 119 mm/joint.
Figure 4: People standing in various poses, captured with a smartphone camera from different viewpoints. Note that this data is significantly different from that obtained from Panoptic, with more challenging outdoor lighting conditions, human-imposed errors from holding and directing the smartphone camera, etcetera. We show two visualization of how Pose-DRL operates in different scenarios. Pose-DRL was not re-trained on this data; we use the same model weights as for producing the results in the main paper. In each scenario we also show the 3d configuration of the scene, as well as which viewpoints are selected by the agent and in which order (pink circles). Left: In this example the agent sees two views before terminating viewpoint selection. The initial randomly given viewpoint produces a pose estimate where the arms are not accurate, which is corrected for in the second and final viewpoint. Right: The agent receives a very good initial viewpoint and decides to terminate viewpoint selection immediately, producing an accurate pose estimate. See § 4.1 for more details about these visualizations.
References


