Supplementary Material: Transforming Self-Supervision in Test Time for Personalizing Human Pose Estimation

Anonymous Author(s)
Affiliation
Address
`email`

1 Pipeline of the Alternative Method

For clarification, we show the alternative method we discussed and compared the proposed method with. It is denoted as *Feat. shared (keypoint)* in Section 4.3. Instead of using a Transformer to model the relation between two sets of keypoints, we simply use a supervised head ψ^{sup} to predict H^{sup}. Two tasks are only connected by sharing a feature backbone.

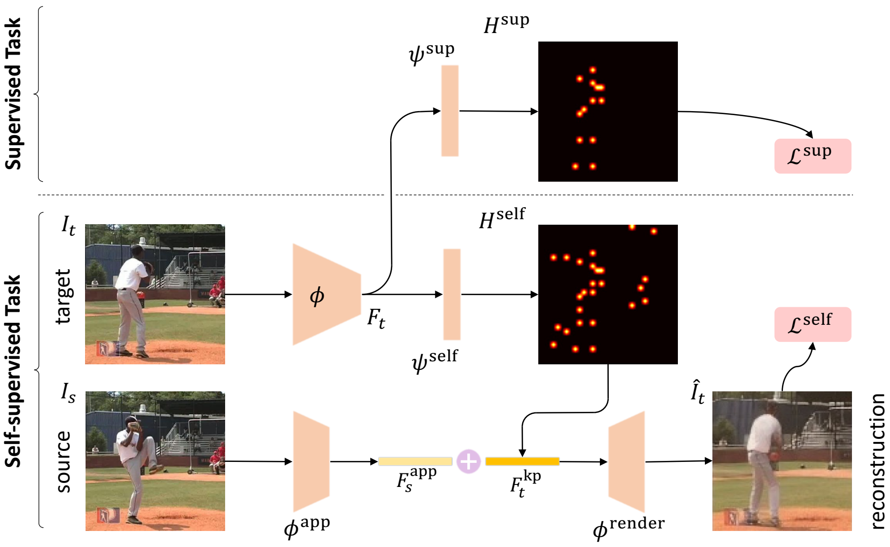

Figure 1: The alternative pipeline *Feat. shared (keypoint)* we discussed and compared the proposed method with.

2 Visualization

In Figure 2 and Figure 3 we visualize our predictions on Penn Action validation set. From top to bottom, the images are: (i) **target image** I_t, i.e. the input image. (ii) **source image** I_s, which provides appearance. (iii) **reconstruction** $\hat{I}_t$. (iv) **self-supervised keypoints**. There are 30 self-supervised keypoints in our setting. (v) **supervised keypoints**. (vi) **ground-truth**.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

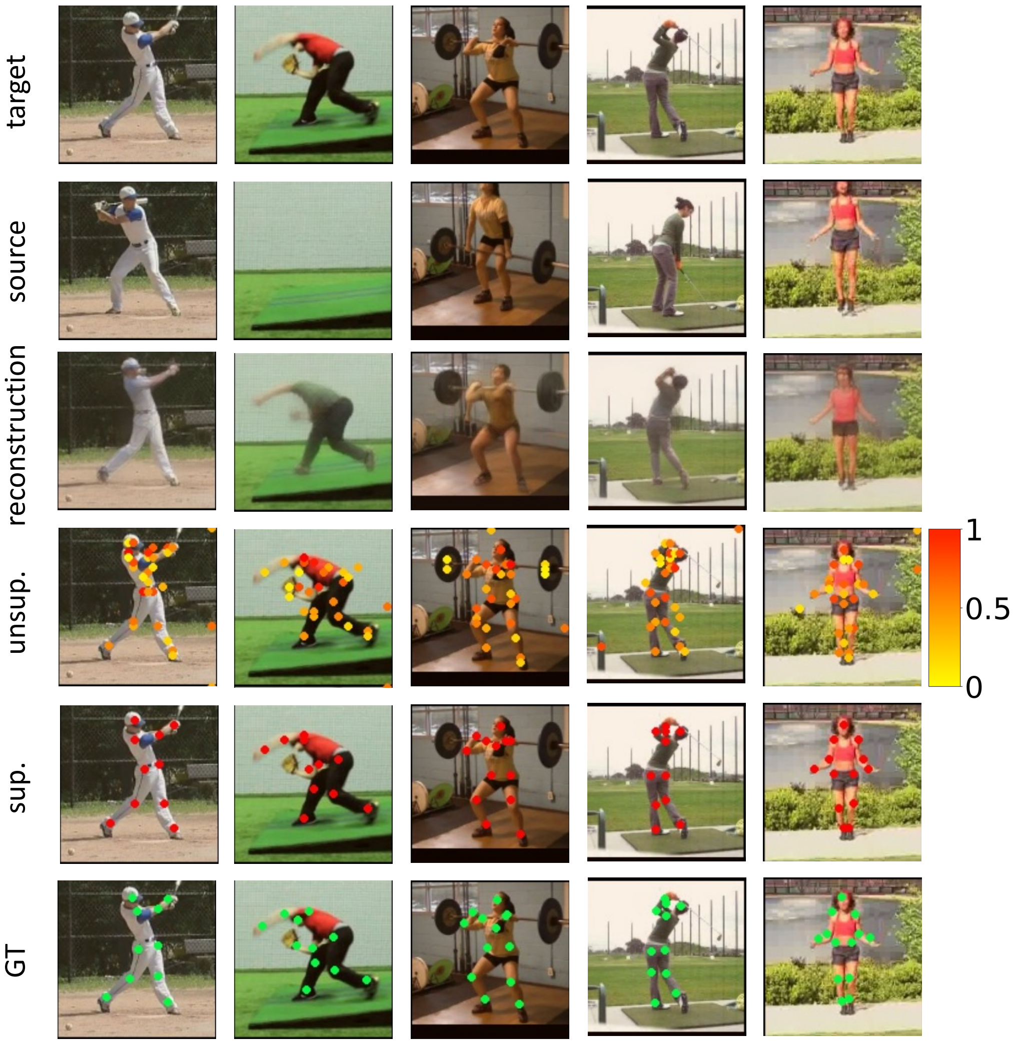

Figure 2: Visualization of our proposed method on Penn Action validation set.

For self-supervised keypoints, we show the contribution of each keypoint to the final pose estimation with color. This is computed as follows. Recall that the Transformer models the relation between two tasks as the affinity matrix

$$W \in \mathbb{R}^{k^{\text{sup}} \times k^{\text{self}}}, \tag{1}$$

where k^{sup} and k^{self} are the number of supervised and self-supervised keypoints. Also recall that

$$H_t^{\text{sup}} = H_t^{\text{self}} W^\top. \tag{2}$$

An entry $W_{i,j}$ actually represents the weight of j-th self-supervised keypoint in computing the i-th supervised keypoint. We then define the contribution of j-th self-supervised keypoint to the final pose prediction as follows

$$c_j = \sum_{i=1}^{k^{\text{sup}}} W_{i,j}. \tag{3}$$

The keypoints with larger c_j are more important to the final pose prediction. Whereas the keypoints with smaller c_j are less important to the final pose prediction and serve to facilitate the self-supervised task of reconstruction.

In Figure 2 and Figure 3 we show the self-supervised keypoints with their contribution to the final pose estimation in the fourth row. It is clear that the points that align with the position of supervised keypoints usually have higher contribution. Other points with deviated positions have

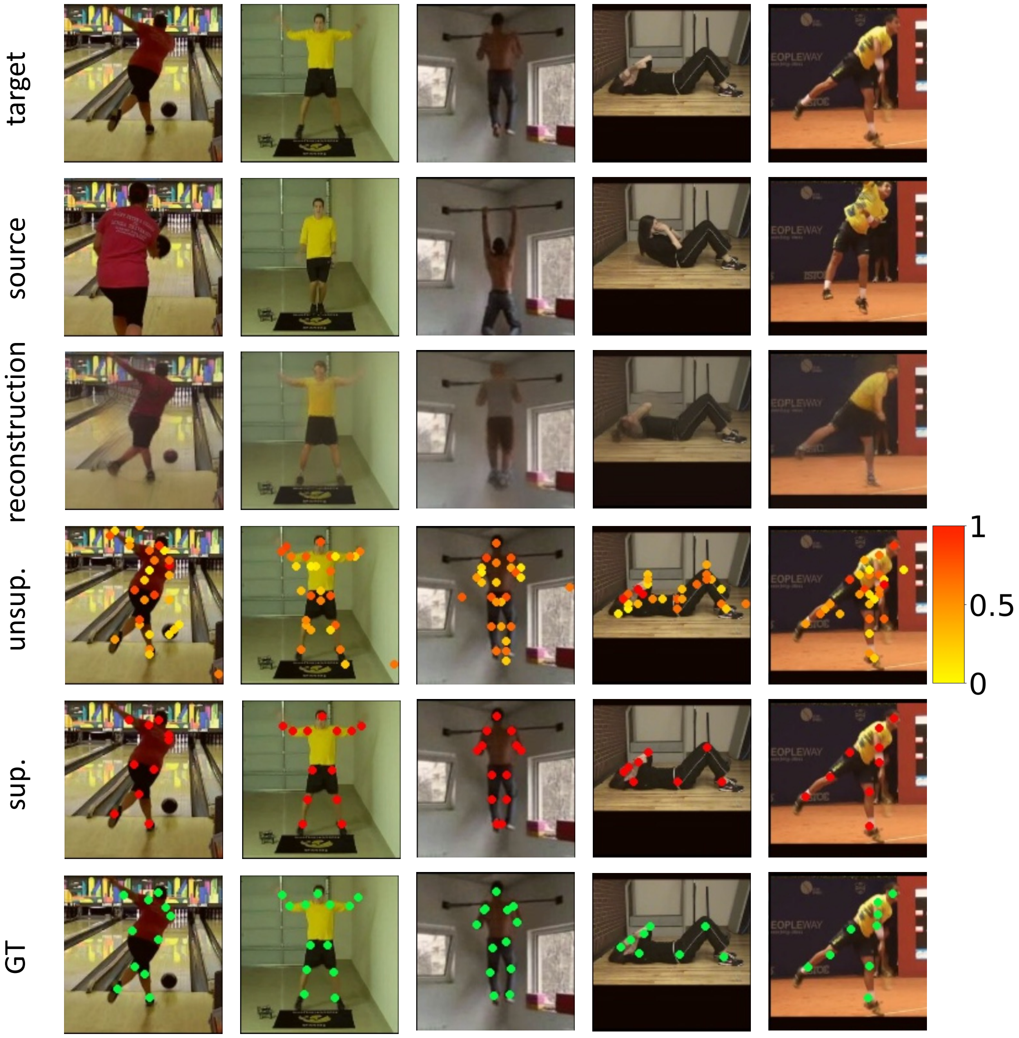

Figure 3: Visualization of our proposed method on Penn Action validation set.

lower contribution. They appear to serve more to facilitate the self-supervised task. Interestingly, some keypoints with minor contribution locate the non-human objects in Penn Action (barbell, bowling ball).

3 More Implementation Details

Batching Strategy. In joint training and TTP offline scenario, both target and source images are randomly chosen and are different within a batch. In TTP online scenario, the target images are always the current frame, which are the same within a batch, whereas the source images are randomly chosen from the previous frames and are different in a batch. In all the situations, each target-source pair is performed data augmentation with same rotation angle and scale factor for the two images to make reconstruction easier. But within a batch, different target-source pairs use different rotation angles and scale factors.

More Model Details. The images I_s and I_t are 128×128. The heatmaps H_t^{sup} and H_t^{self} are 32×32. In self-supervised task, appearance information F_s^{app} and keypoint information F_t^{kp} has size 16×16 with 256 channels. For the perceptual loss, we use a VGG-16 network pretrained on ImageNet to extract semantic informations. We do not use flip test during inference.

All our models are trained on a single Tesla V100 GPU. For more details, please refer to the code in the supplementary materials.