Supplementary: Pose-Transformation and Radial Distance Clustering for Unsupervised Person Re-identification

Siddharth Seth
ssiddharth@iisc.ac.in
Akash Sonth
akashsonth@vt.edu
Anirban Chakraborty
anirban@iisc.ac.in

Visual Computing Lab,
Department of Computational and Data Sciences,
Indian Institute of Science,
Bangalore, India

We perform additional analysis of the proposed unsupervised learning framework in this supplementary. The sections are organized as follows:

- Section 1: Notations
- Section 2: Pose-transformed dataset
- Section 3: Overall training algorithm
- Section 4: Experimental analysis

1 Notations

We provide a description for the notations used in the main paper and supplementary in Table 1 to support easier referencing.

2 Pose-transformed dataset

We describe in detail here about the parts transformation sections mentioned in the main paper. We randomly sample poses from the re-ID dataset itself to ensure that the generated samples follow a re-ID setup.

2.1 Parts transformation

The pose and parts of an image, x_i^o are transformed to generate the image, x_j^p. We next describe the process of spatial transformation, $\mathcal{S}_t$ for a randomly sampled target pose, p_t^o. Fig. 1 gives an overview of the transformation process. The input to $\mathcal{S}_t$ are the source pose, p_i^o, the corresponding body parts, b_i^o (both extracted from the image, x_i^o), and the target

Table 1: Notations Table

Symbol	Description
$\mathcal{F}$	Network to be trained
$x_i^o, y_i^o \in X_o$	Original dataset with pseudo-labels
N	Number of samples in original dataset
$x_j^p \in X_p$	Pose-transformed dataset
K	Constant scaling factor for PT dataset
$x_i, y_i \in \mathbb{D}$	Pseudo-labeled training set consisting of both original and PT dataset
Z	Labels corresponding to $\mathbb{D}$
p_i^o	Extracted poses from images x_i^o
p_t^o	Randomly sampled target poses
b_i^o	Extracted body parts from images x_i^o
$\mathcal{S}_t$	Spatial transformation function for target pose p_t^o
$\mathcal{C}$	Number of clusters for hierarchical clustering
C_l	Cluster with index l
r_l	Radius corresponding to cluster C_l
c_l	Centroid corresponding to cluster C_l
d_i^l	Distance between sample x_i^o and cluster C_l
γ	minimum radial distance

random pose, p_t^o to which the body parts need to be transformed. All the 3D poses are projected to 2D via a random camera transformation before being utilized in the spatial transformation function.

Each body part $b_i^o(m)_{m=1}^{10} \in \mathbb{R}^{H\times W\times 3}$, except the torso, is represented using 2 joints. Let the two joints in 2D, forming a part $b_i^o(m)$ be denoted as $J_1(m)$ and $J_2(m)$. Next, let $R_1(m)$ and $R_2(m)$ be the two joints representing the same part in the 2D projection of the target random pose p_t^o. After aligning all body parts of p_i^o along the $+ve$ X-axis, the vector from mid-point, $(J_1(m)+J_2(m))/2$, of part $b_i^o(m)$ to $J_2(m)$ makes an angle of 0° wrt the $+ve$ X-axis. The part after aligning with the X-axis is centered at the origin (0,0).

The rotation parameter, α between the original part orientation (with joints $J_1(m)$ and $J_2(m)$) and the target part orientation (with joints $R_1(m)$ and $R_2(m)$) is calculated as

$$\alpha(m) = -\text{atan2}((R_2^y(m) - R_1^y(m)), (R_2^x(m) - R_1^x(m))) \tag{1}$$

where x and y in subscripts represent the X and Y coordinates of the joints R_1 and R_2 in the image space.

The next transformation involves scaling of the part $b_i^o(m)$ according to target part length. We assume the part width remains fixed as modeling width would require an additional pair of joints. The scaling transformation is then defined as

$$\beta_x(m) = dist(J_1(m), J_2(m))/dist(R_1(m), R_2(m)) \tag{2}$$

$$\beta_y(m) = 1 \tag{3}$$

where $dist(.,.)$ denotes the Euclidean distance between two joints in image space.

The last step requires translating the part $b_i^o(m)$ to the randomly chosen target pose's corresponding part location. One can simply find the mid-point of the target pose's part as

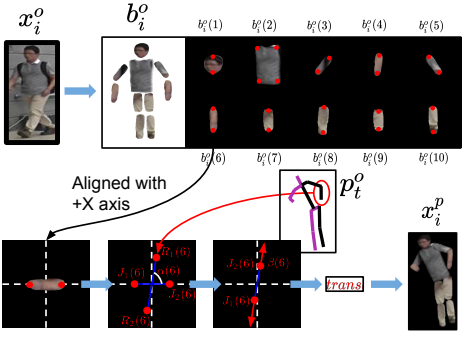

Figure 1: A sample illustration of spatially transformed body parts.

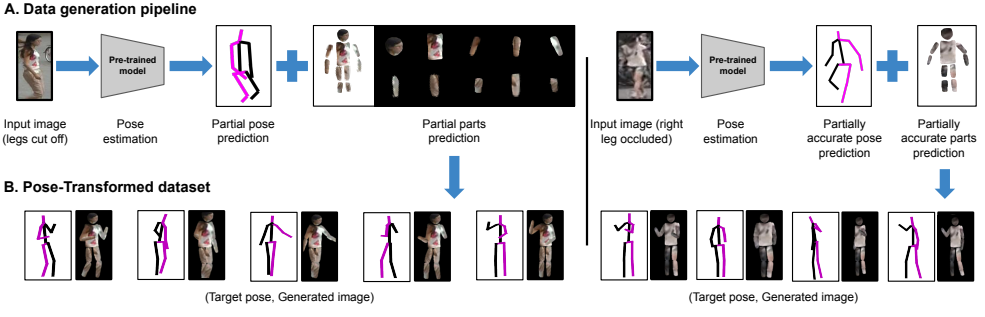

Figure 2: Challenges with PT dataset. The left panel shows how spatial transformation caters to partial occlusion or truncation. The right panel shows noisy parts prediction may lead to noisy spatial transformations.

$(R_1^x(m)+R_2^x(m))/2$ and $(R_1^y(m)+R_2^y(m))/2$ for X and Y coordinates in the image space and translate the origin to this location.

Finally, the torso is represented by 4 joints, two at shoulders and two at the hips. This gives us the ability to shrink and expand the torso in both X and Y directions. This is specifically helpful in cases when the source pose is front facing (where the torso is expanded) and needs to be transformed to a side pose (where the torso is shrunk). Thus, only for torso, scaling occurs across both X and Y directions. The line joining the mid-points of the hips and shoulders is considered for computing the rotation parameter α.

2.2 Discussion

The PT dataset already comprises both natural and unnatural cases. We do not in any way separate the unnatural-looking images, making PT dataset easier to adapt. The appearance does not vary significantly, and thus, the overall performance remains almost the same even when some unnatural PT images are present in the dataset. On ten runs, we obtain a mean rank-1 performance of 93.6 with a standard deviation of 0.2, and a mean of 81.6 with an approximate standard deviation of 0.15. Sec. 4 compares PT dataset with other image generation techniques.

Fig. 2 shows two examples of the samples generated from X_o. The left one shows how the spatial transformation effectively caters to body parts cut off from the image. The legs here are not fully captured in the image. Thus, the pose estimation model truncates the legs and the corresponding parts after the knee joints. However, due to the scaling operation during spatial transformation, the truncated part is stretched to the target part's pose.

The second example, on the right, depicts the case when incorrect pose prediction leads to noisy parts prediction which further leads to noisy parts transformation. Here, for example, the right leg is incorrectly predicted on the background. The parts thus obtained contain background information. Any transformation of these will lead to all transformed images containing background in the right leg. This limits us in generating the number of samples per identity (scaling factor K). However, such aberrations do not adversely affect the model as the performance peaks at $K = 25$ but saturates after that and does not noticeably degrade (Fig. 3, main paper).

Null background. We aim to disentangle the factors of variation across the images of a particular identity, i.e., pose and background, where the appearance remains constant. This will let the model focus on the only important cue for re-ID, i.e., foreground (person) appearance. We disentangle pose by generating multiple images of the same person in different poses. To recognize all these images belonging to the same identity, the model discards the only factor of variation, i.e., pose. Similarly, using a null background in PT data lets the model only focus on the foreground person. We still use the original dataset, which helps the model adapt to images containing different backgrounds. Table 2 compares methods that generate backgrounds but lack overall generation quality. Briefly, foreground appearance plays the most crucial role for re-ID and is thus the center of focus in the PT dataset.

Train $\mathcal{F}$ on $\mathbb{D}$ to initialize its parameters.
for *each epoch* **do**
  Extract features $\mathcal{F}(X_o)$
  Assign each image x_o to an individual cluster.
  for *each step s* **do**
    Calculate distance between each cluster according to [2].
    Merge m nearest clusters.
  end
  Assign pseudo labels to X_o as per the obtained $\mathcal{C}$ clusters.
  Get centroids c_l and radii r_l for every cluster C_l.
  for *each iteration* **do**
    for *each sample* x_i^o *in mini-batch* **do**
      Compute $\mathcal{L}_{cls}$ for x_i^o.
      Compute $\mathcal{L}_{tri}$ for x_i^o, x_{i+}^o, and x_{i-}^o.
      Compute $\mathcal{L}_{sd}$ for x_i^o.
    end
    Optimize $\mathcal{F}$ over $\mathcal{L}_{cls}$, $\mathcal{L}_{tri}$, and $\mathcal{L}_{sd}$ using Adam optimizer
  end
end

Algorithm 1: Training algorithm for the proposed discriminative learning framework.

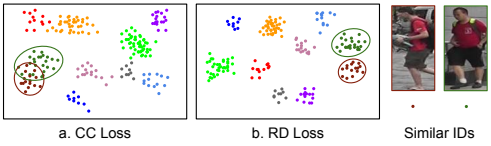

Figure 3: t-SNE visualization for **a.** Contrastive-Center loss and **b.** Radial distance loss for 10 different identities. 2 identities (red, green) are shown as similar-looking.

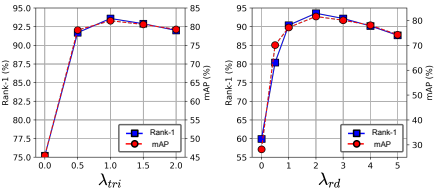

Figure 4: Ablation on Market-1501 by varying λ_{tri} and λ_{rd}.

3 Overall training algorithm

Algorithm 1 summarizes the complete training pipeline of the proposed unsupervised learning framework. The first step involves training $\mathcal{F}$ on the pseudo-labeled dataset $\mathbb{D}$. We term this as the *Initialization Step*. This is followed by the discriminative clustering pipeline. We use the hierarchical clustering algorithm [3] to classify X_o in $\mathcal{C}$ clusters. After calculating the radii and centroids for all the clusters, we apply the radial distance loss on the dataset to form compact clusters with significant inter-cluster distances. The discriminative step significantly boosts the discriminative ability of the network $\mathcal{F}$. We demonstrate this through extensive experimental analysis, both in the main paper and the supplementary.

4 Experimental analysis

We describe additional experimental analysis of the proposed framework in this section. We use $\tau = 0.1$, $\gamma = 0.5$, $\lambda_{tri} = 1$, and $\lambda_{rd} = 2$ for our best reported results. We employ the hierarchical clustering algorithm along with the best values of hyperparameters used such as $\mathcal{C}$, m and s from [3] for all our reported results.

Comparison to contrastive-center loss. CCL [1] aims to maximize the inter-cluster distances but a strict separability is not guaranteed. Fig. 3 shows clusters learned on the training data. In CCL, points from different clusters overlap even for dissimilar identities. On the other hand, RD loss maintains a strict separability thereby separating even similar-looking identities. Using CCL on Market-1501 yields 78.5@R1 vs 93.6@R1 (Ours).

Ablation on hyperparameters. Fig. 4 shows an ablation on the hyperparameters used in Eq.4. We obtain best results at $\lambda_{tri} = 1$ and $\lambda_{rd} = 2$. The hyperparameters are robust to small change in values. Significantly weighing λ_{rd} reduces the effect of triplet loss which plays a role in moving similar features closer and dissimilar features farther. Significantly weighing λ_{tri} reduces the effect of the radial distance loss.

References

[1] Ce Qi and Fei Su. Contrastive-center loss for deep neural networks. In *ICIP*, 2017.

[2] Robert R Sokal. A statistical method for evaluating systematic relationships. *Univ. Kansas, Sci. Bull.*, 1958.

[3] Kaiwei Zeng, Munan Ning, Wang Yaohua, and Yang Guo. Hierarchical clustering with hard-batch triplet loss for person re-identification. In *CVPR*, 2019.