Pose-Guided Self-Training with Two-Stage Clustering for Unsupervised Landmark Discovery

Supplementary Material

7. Pseudo-code D-ULD++

The architecture for D-ULD++ is shown in Fig. 2 main manuscript. The input to the architecture is an image $\mathbf{x}_j$. The aggregator network Ψ_b branches into the descriptor head and the detector head with the VAE auto-encoder appended to it. The output of the descriptor head is given by the following sequence of operations $\mathbf{F}^j = \Psi_f(\Psi_b(\mathbf{x}_j))$. The operations for the modified detector head is given by $\boldsymbol{l}_j = \Psi_V^{Enc}(\Psi_d(\Psi_b(\mathbf{x}_j)))$.

The following contrastive loss is minimized for the descriptor head.

$$\mathcal{L}_{\mathbf{f}}(\mathbf{f}_i^j, \mathbf{f}_{i'}^{j'}) = \mathbf{1}_{[\mathbf{c}_i^j = \mathbf{c}_{i'}^{j'}]} ||\mathbf{f}_i^j - \mathbf{f}_{i'}^{j'}|| + \mathbf{1}_{[\mathbf{c}_i^j \neq \mathbf{c}_{i''}^{j''}]} \max(0, m - ||\mathbf{f}_i^j - \mathbf{f}_{i''}^{j''}||) \tag{4}$$

Descriptors with the same labels $\mathbf{c}_i^j = \mathbf{c}_{i'}^{j'}$ are pushed together, whereas those with different are minimized unless separated by a margin m.

Likewise for the detector head, we minimize the following loss:

$$\mathcal{L}_{\boldsymbol{\varphi}}(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_{j'}) = \mathbf{1}_{[\boldsymbol{u}_j = \boldsymbol{u}_{j'}]} ||\boldsymbol{\varphi}_j - \boldsymbol{\varphi}_{j'}|| + \mathbf{1}_{[\boldsymbol{u}_j \neq \boldsymbol{u}_{j''}]} \max(0, m - ||\boldsymbol{\varphi}_j - \boldsymbol{\varphi}_{j''}||) \tag{5}$$

Equation (5) pushes latent codes with the same labels together, *i.e.* $\boldsymbol{u}_j = \boldsymbol{u}_{j'}$.

The pseudo-code for D-ULD++ is described in Algorithm 2.

8. Consistency Analysis

We perform consistency analysis to evaluate whether the detected landmarks are consistent or not [42]. The consistency of detected landmarks is defined as, $e_k = ||\Psi_d(\Psi_b(A(\mathbf{x}_j))) - A(\Psi_d(\Psi_b(\mathbf{x}_j)))||)$, where A is a random similarity transformation. Ψ_d and Ψ_b are the descriptor head and aggregator network respectively.

We report consistency errors, averaged over $K = 10$ landmarks, in Table 4. Our method produces more consistent landmarks than the competing approaches on all datasets.

Method	MAFL	AFLW	CatHeads	LS3D
Sanchez [42]	8.78	7.56	2.58	21.3
Awan [2]	2.37	1.77	2.24	3.23
D-ULD++ (Ours)	1.56	0.87	1.78	1.98

Table 4. Our method (D-ULD++) produces more consistent landmarks than the competing methods across all datasets.

9. Additional CED Curves

Figure 9 shows the cumulative error curves (CED) curves for CatHeads and AFLW datasets. In concurrence with the CED curves from the main manuscript, our method shows significantly lower base error and a more gradual degredation in performance.

10. Qualitative Results

We show additional qualitative results for LS3D (Figure 12), CatHeads (Figure 11) and AFLW (Figure 10) comparing 3 methods, Jakab, Mallis and D-ULD++. Jakab [17] generally learns landmarks with poor localization, occasionally not even lying in the image ROI. Mallis [32] performs much better localizing most landmarks well, but a few landmarks are still in smooth regions that lack distinctive edges and are thus poorly localized. Finally, D-ULD++ is reliably able to localize landmarks that are lying in image regions with distinctive edges.

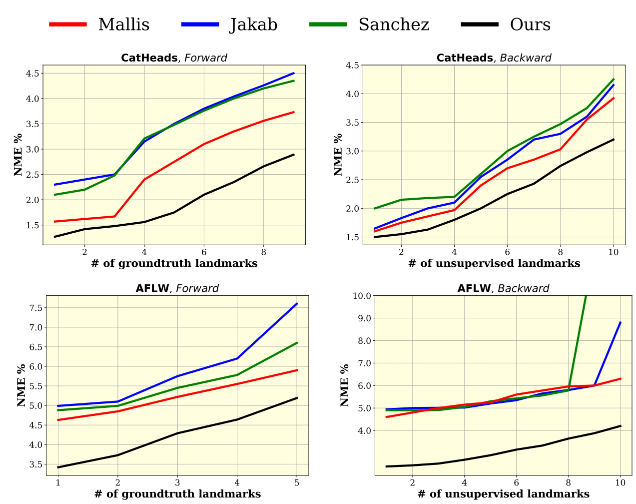

Figure 9. Cumulative Error Distribution (CED) Curves of forward and backward NME for CatHeads and AFLW.

Algorithm 1 Update-Dataset $\mathcal{X}$

Input: $\mathcal{X} = \{\mathbf{x}_j \mid j \in \text{images}\}$

1. $\{\mathbf{p}_i^j, \mathbf{f}_i^j\}_{i \in N_j} =$ Extract keypoints and descriptors from $\Psi(\mathbf{x}_j)$ ▷ Keypoints and descriptors are extracted for each image $\mathbf{x}_j$.
2. $\mathcal{X} = \{\mathbf{x}_j, \{\mathbf{p}_i^j, \mathbf{f}_i^j, \mathbf{c}_i^j\}_{i=1}^N\}$ ▷ Update $\mathcal{X}$ with keypoints, descriptors and cluster pseudo-labels. $\{\mathbf{p}_i^j, \mathbf{f}_i^j, \mathbf{c}_i^j\}_{i \in N_j}$.
3. $\boldsymbol{\varphi}_j = \Psi_V^{Enc}(\Psi_d(\Psi_b(\mathbf{x}_j)))$ ▷ Extract the latent codes for each image $\mathbf{x}_j$.
4. $\boldsymbol{l}_j$ = KMeans($\{\boldsymbol{\varphi}_j\}$) ▷ Compute pose latent-code cluster labels $\boldsymbol{l}_j$.

Output: $\mathcal{X} = \{\mathbf{x}_j, \{\mathbf{p}_i^j, \mathbf{f}_i^j, \mathbf{c}_i^j\}, \boldsymbol{l}_j, \boldsymbol{u}_j\}$.

Algorithm 2 Pseudo-Code D-ULD++

$\mathcal{X}$ =Update-Dataset($\mathcal{X}$) ▷ $\mathcal{X}$ is updated. $\mathcal{X} = \{\mathbf{x}_j, \{\mathbf{p}_i^j, \mathbf{f}_i^j, \mathbf{c}_i^j\}, \boldsymbol{l}_j, \boldsymbol{u}_j\}$.
Main Training Loop
for epoch = $1 \rightarrow N_E$ **do** ▷ Epoch loop.
  for $i = 1 \rightarrow N_{it}$ **do** ▷ Iterate for N_{it} iterations.
    $\{\mathbf{x}_j, \{\mathbf{p}_i^j, \mathbf{f}_i^j, \mathbf{c}_i^j\}, \boldsymbol{l}_j, \boldsymbol{\varphi}_j\}$ = GetBatch($\mathbf{x}_j$)
    Update the network Ψ, Ψ_V^{Enc} with the gradients of$\mathcal{L}_f$ and $\mathcal{L}_{\boldsymbol{\varphi}}$.
  end for
  5. Re-populate $\mathcal{X}$ by redoing steps 1 to 4. $\mathcal{X} = \{\mathbf{x}_j, \{\mathbf{p}_i^j, \mathbf{f}_i^j, \mathbf{c}_i^j\}, \boldsymbol{l}_j, \boldsymbol{u}_j\}$
end for

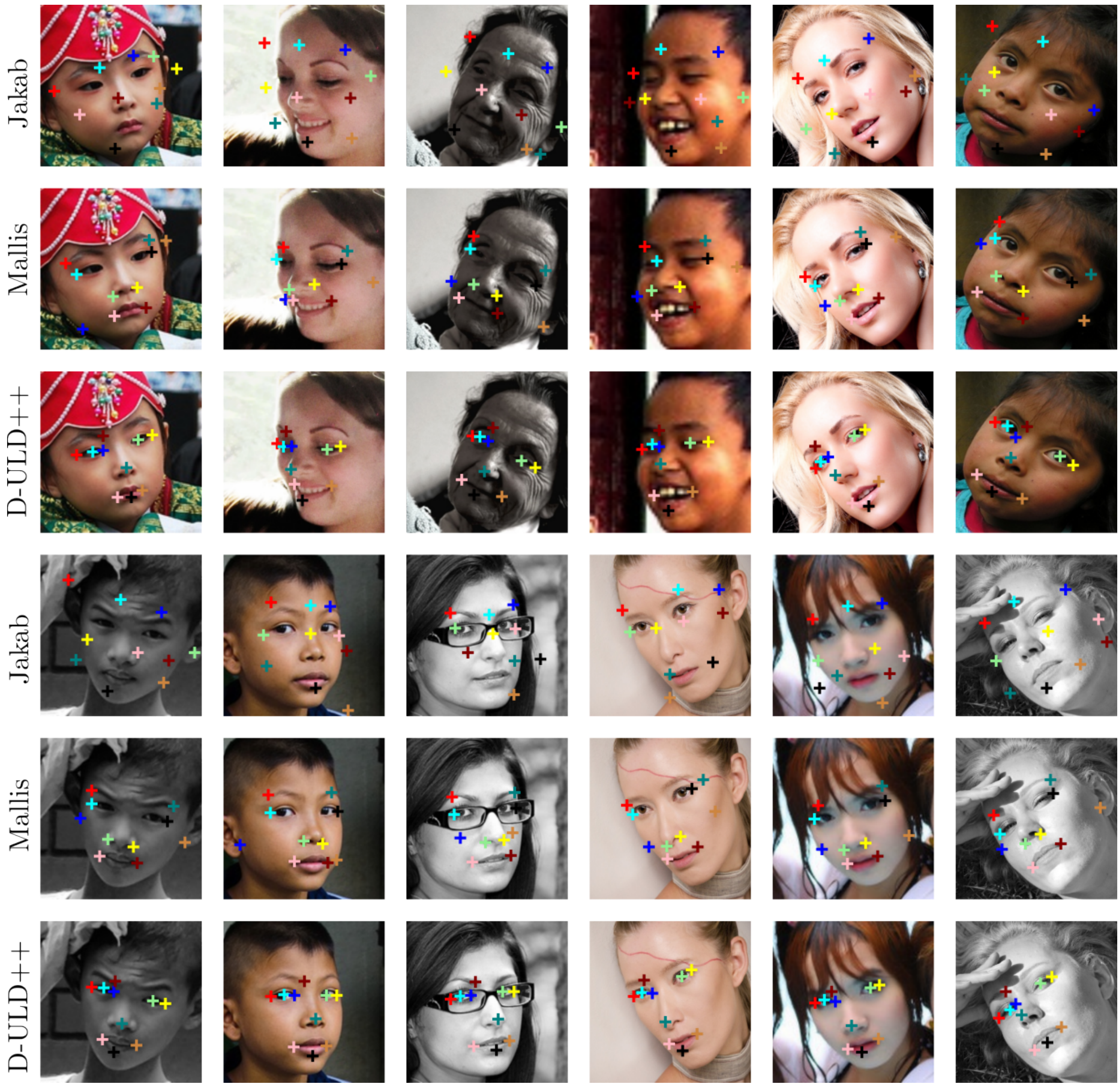

Figure 10. Results Comparison on AFLW for Jakab, Mallis and D-ULD++.

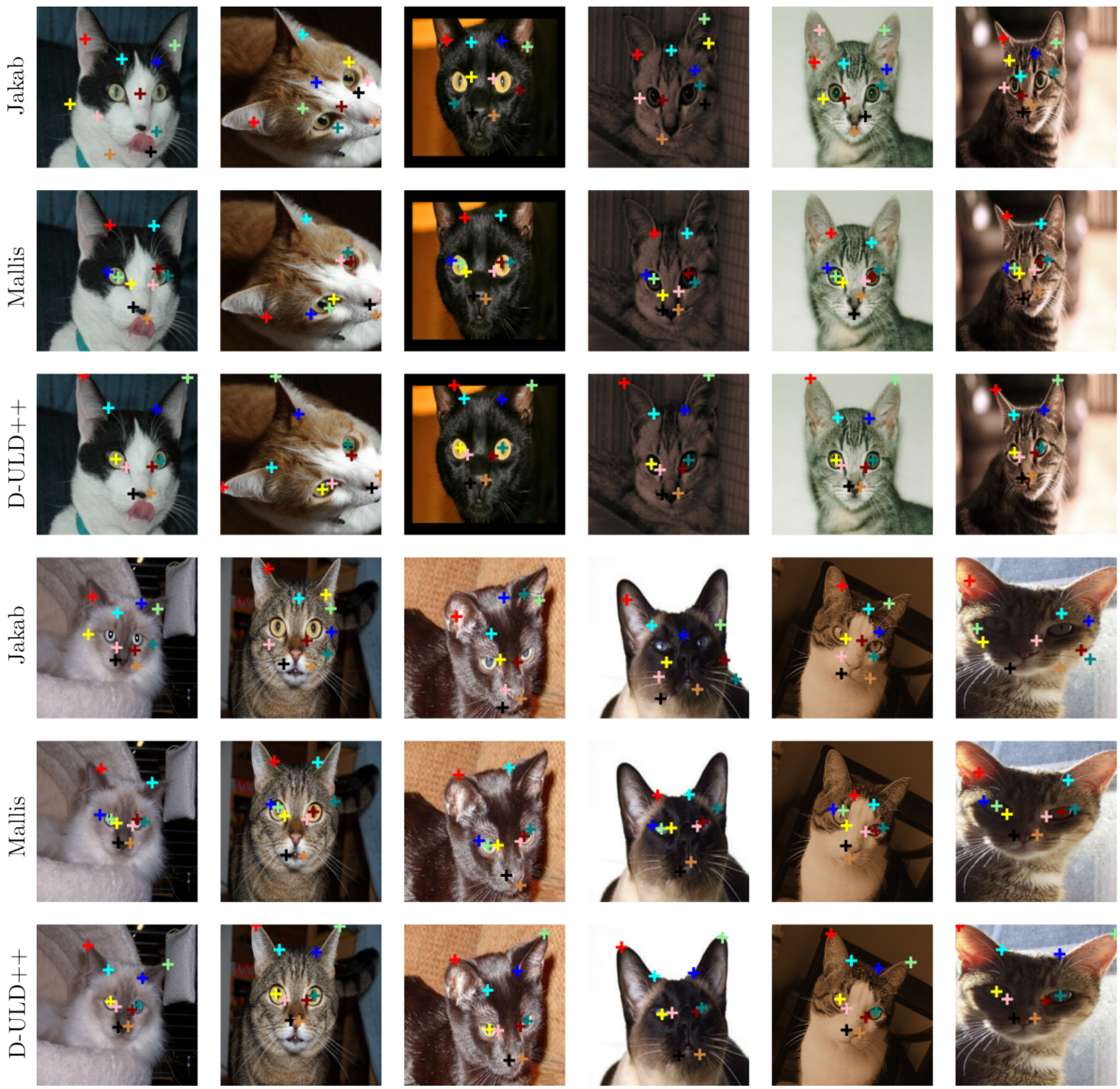

Figure 11. Results Comparison on CatHeads for Jakab, Mallis and D-ULD++.

Figure 12. Results Comparison on LS3D for Jakab, Mallis and D-ULD++.