PoseBH: Prototypical Multi-Dataset Training Beyond Human Pose Estimation Supplemental Document

Uyoung Jeong[1] Jonathan Freer[2] Seungryul Baek[1] Hyung Jin Chang[2] Kwang In Kim[3]
[1]UNIST [2]University of Birmingham [3]POSTECH

In this supplementary document, we provide details on the keypoint embedding module (Sec. 1) and experimental settings (Sec. 2), as well as additional experimental results Sec. 3).

1. Keypoint embedding module details

Our keypoint embedding module consists of two upsampling layers, one residual block, and two convolution layers. For the upsampling layers, we adopt the head design from ViTPose++ [7]. The residual block combines a standard 3×3 convolution with a skip connection, where we replace the ReLU activation with SiLU [1]. The final two convolution layers include a 3×3 convolution, batch normalization, SiLU activation, and a 1×1 output convolution. Although lightweight and simple, our embedding module effectively enhances multi-dataset training. We initialize the prototypes using a truncated normal distribution with a mean of 0, a standard deviation of 0.02, and a range of $[-2, 2]$, followed by L2-normalization across the embedding dimension.

2. Experimental setup

2.1. Hyperparameters

We set the embedding dimension to $F = 64$, the number of in-class prototypes to $M = 3$, and the total number of keypoints to $J = 214$. The output heatmap dimensions were set to a width of $W = 48$ and a height of $H = 64$. Following [11], we set $\kappa = 0.05$ for obtaining the target t_j. The prototype update momentum λ was set to 0.999. For the loss weights, we set $\alpha = 3.33 \times 10^{-6}$, $\beta = 1.25 \times 10^{-7}$, $\gamma = 1.25 \times 10^{-8}$, $\delta = 0.01$, and $\zeta = 0.001$. The impact of varying hyperparameter values (on the COCO validation set, measured in mean average precision; AP) is presented in Tabs. 1 to 3, demonstrating robust performance across different hyperparameter settings. The final hyperparameter values are highlighted in bold.

F	AP
32	77.1
64	77.1
128	77.1

Table 1. Impact of varying the embedding dimension F (mean AP on COCO validation set).

α	AP
$\mathbf{3.33 \times 10^{-6}}$	77.1
6.25×10^{-6}	77.1
1.25×10^{-5}	77.1

β	γ	AP
1.00×10^{-7}	1.00×10^{-8}	77.1
$\mathbf{1.25 \times 10^{-7}}$	$\mathbf{1.25 \times 10^{-8}}$	77.1
5.00×10^{-7}	5.00×10^{-8}	77.1
1.00×10^{-6}	1.00×10^{-7}	77.1

Table 2. Impact of varying the loss weight values α, β, and γ (mean AP on COCO validation set).

δ	AP	ζ	AP
5.0×10^{-3}	77.3	1.0×10^{-4}	77.3
$\mathbf{1.0 \times 10^{-2}}$	77.2	$\mathbf{1.0 \times 10^{-3}}$	77.2
5.0×10^{-2}	77.3	1.0×10^{-2}	77.3

Table 3. Impact of varying the loss weight values δ and ζ (mean AP on COCO validation set).

2.2. APT-36K preprocessing

For APT-36K [9], since official train, validation, and test splits are not provided, we partitioned the dataset using a 7:1:2 ratio following the guidelines of the original paper. This resulted in approximately 24,900 images and 37,000

Method	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
ViTPose++-B	76.4	92.7	84.3	73.2	82.2	81.5
ViTPose++-H	78.5	**93.4**	**86.2**	**75.3**	**84.4**	83.4
Ours-B	76.6	92.6	84.4	73.4	82.4	81.7
Ours-H	**78.6**	93.3	**86.2**	**75.3**	**84.4**	**83.5**

Table 4. COCO test-dev evaluation results.

Method	Val	Test	Val (occ)	Test (occ)
ViTPose++-B	81.1	82.0	64.0	64.1
ViTPose++-H	85.7	86.8	72.6	72.9
Ours-B	82.2	83.1	66.3	66.2
Ours-H	**86.0**	**87.0**	**73.2**	**73.7**

Table 5. OCHuman evaluation results (measured in mean average precision; AP). Ground-truth bounding boxes were used for cropping.

Method	Average score
ViTPose++	68.2
+UniDet	68.9
Ours	**70.6**

Table 6. Performance of different MDT methods (average over six datasets).

Method	InterHand	3DPW
ViTPose++	86.2	81.7
Trn. scratch	86.1	56.8
AIC-trained	86.3	81.5
Ours	**87.1**	**83.6**

Table 7. Performance of domain transfer methods on the InterHand2.6M (AUC) and 3DPW (AP) datasets.

instances for training, 3,600 images and 5,400 instances for validation, and 6,900 images and 10,700 instances in testing. To ensure that videos in the validation and test sets do not appear in the training set, each video is assigned to a single split.

2.3. 3DPW

We use the processed annotations from ScoreHypo [8]. To obtain 2D keypoint annotations, we use the 2D projected 3D SMPL keypoints. We reformat the annotations to the COCO style and employ COCO-style evaluation metrics.

2.4. Training.

We trained our method on a system with four NVIDIA A100 GPUs or four NVIDIA A6000 GPUs using PyTorch 1.11 in an Ubuntu 20 environment. To enhance training efficiency, we also enabled automatic mixed precision with distributed training. We set the random seed value to 0 for all experiments to avoid randomness during training.

For learning rate scheduling, we start with an initial learning rate of 0.001 and reduce it by a factor of 0.1 at the 50th and 90th epochs. During the first 50 epochs, only the embedding module and prototypes are updated, while all other components remain frozen. At the start of the 50-th epoch, we set the backbone and the multi-heads to be trainable and freeze the prototypes. We then apply the $\mathcal{L}_{\text{CSS}}$ loss function (Eq. 8 in the main paper).

For transfer learning on InterHand2.6M, we follow ViTPose++ configurations. We train the model for 60 epochs with 5.0e-4 initial learning rate. In the case of transfer learning on 3DPW, we train the model for 30 epochs with 1.0e-4 initial learning rate. We train the prototypes for 30 epochs in InterHand2.6M, and 15 epochs in 3DPW.

3. Additional results

3.1. Quantitative results

Table 4 presents the pose estimation results on the COCO test-dev set. Following previous works [3, 5, 6], we cropped the input images based on detected bounding boxes. Our method outperforms the baseline ViTPose++ by 0.2 AP with the *Base* backbone and by 0.1 AP with the *Huge* backbone.

We also performed a downstream evaluation on unseen data using the OCHuman [10] dataset, which comprises 2,500 validation images and 2,231 test images, with no available training set. The results are presented in Tab. 5. Since OCHuman follows the COCO keypoint format, we used COCO-trained models for evaluation. Here, '(occ)' denotes the evaluation of the occluded keypoints, following the protocol of [2]. Our method outperforms the baseline ViTPose++, particularly on the (occ) subsets, demonstrating its robustness to occlusion.

Pose estimation lacks a large, generic, high-quality dataset, which is a key motivation for our multi-dataset training (MDT) approach. As shown in Tab. 7, our method outperforms models trained from scratch or transferred from the *largest* single dataset (AIC) in domain transfer scenarios for pose estimation. Furthermore, compared to existing MDT problems (e.g., classification and detection), skeletal heterogeneity in pose estimation presents a unique challenge, making naïve dataset merging or multi-head supervision ineffective. Table 6 demonstrate this: Our method significantly outperforms conventional label merging approaches when applied to pose datasets such as UniDet [12].

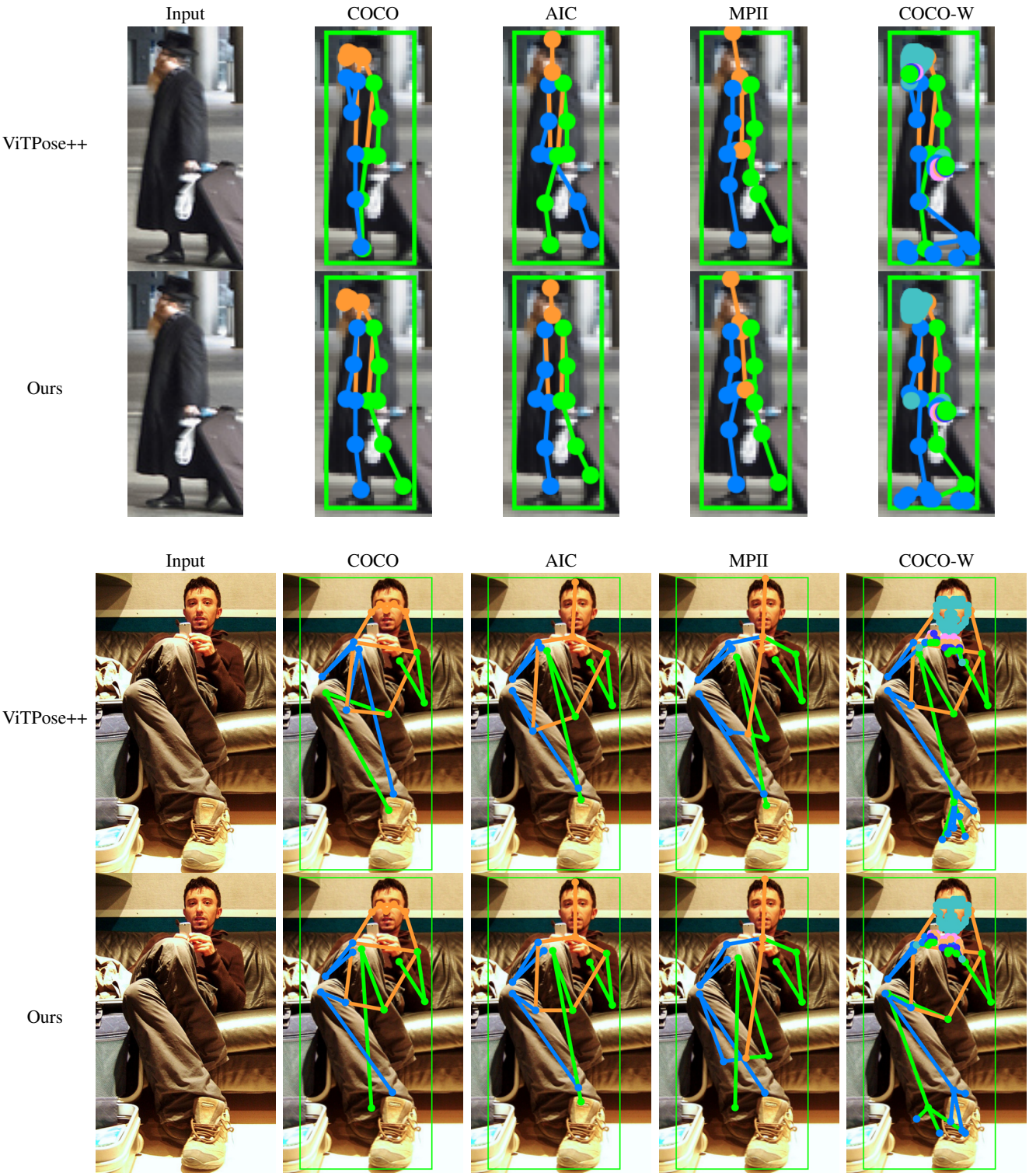

Figure 1. Pose estimation examples comparing ViTPose++ and our method.

3.2. Qualitative results

Figure 1 presents additional human pose estimation examples. In the first two rows, ViTPose++ struggles with accu-

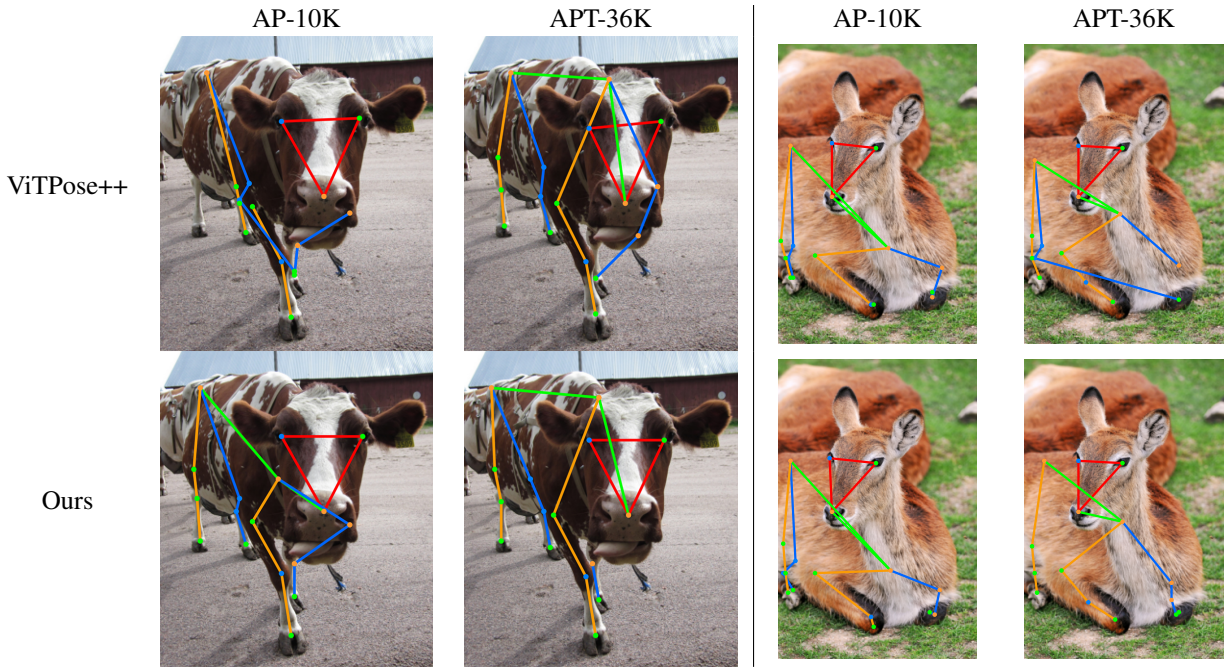

Figure 2. Pose estimation examples for animals using ViTPose++ and our method.

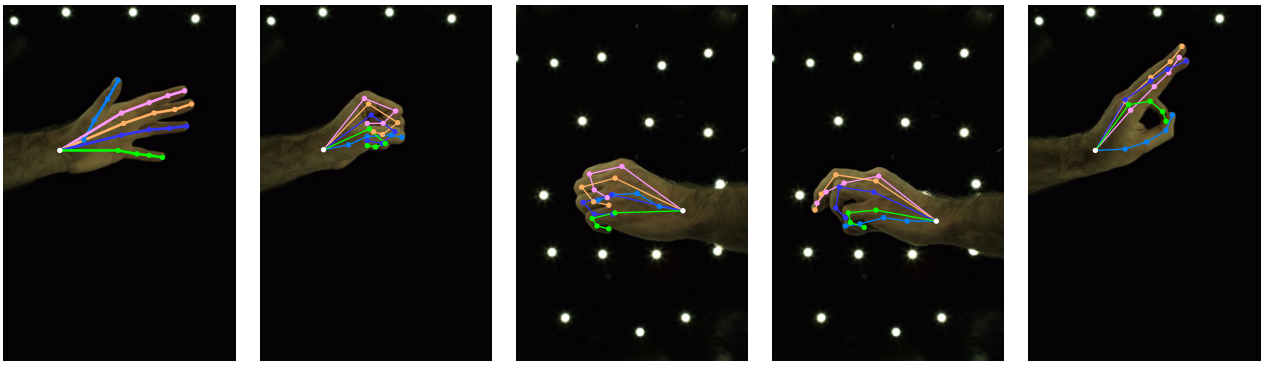

Figure 3. Pose estimation examples from our algorithm on the InterHand2.6M dataset.

rate leg estimation and exhibits inconsistent keypoint predictions across different dataset skeletons. In contrast, our method accurately estimates the legs and maintains consistency across varying skeletons. In the bottom two rows, ViTPose++ incorrectly predicts the right and left foot at the same location, whereas our method correctly estimates the legs, except for AIC.

Figure 2 provides additional comparisons on the AP-10K animal dataset. For cattle (first two columns), ViTPose++ mislocalizes the left front leg in the APT-36K skeleton, while our method correctly identifies it. Similarly, in the last two columns (kangaroo), ViTPose++ confuses the right front leg with the left, an error our method successfully avoids.

Figure 3 and Figure 4 provide additional pose estimation examples of our algorithm on the InterHand2.6M and 3DPW datasets, respectively. Our method demonstrates strong generalization across hand and human shapes, even under various self-occlusion and external occlusion scenarios.

We visualize the prototypes constructed by our algorithm in Fig. 5, using those trained with the ViT-B backbone. The InterHand and 3DPW prototypes are separately trained during the domain transfer process, while others are jointly learned during MDT. The prototypes effectively capture the diversity of representations within the embedding space and successfully align similar keypoints across different datasets without compromising localization perfor-

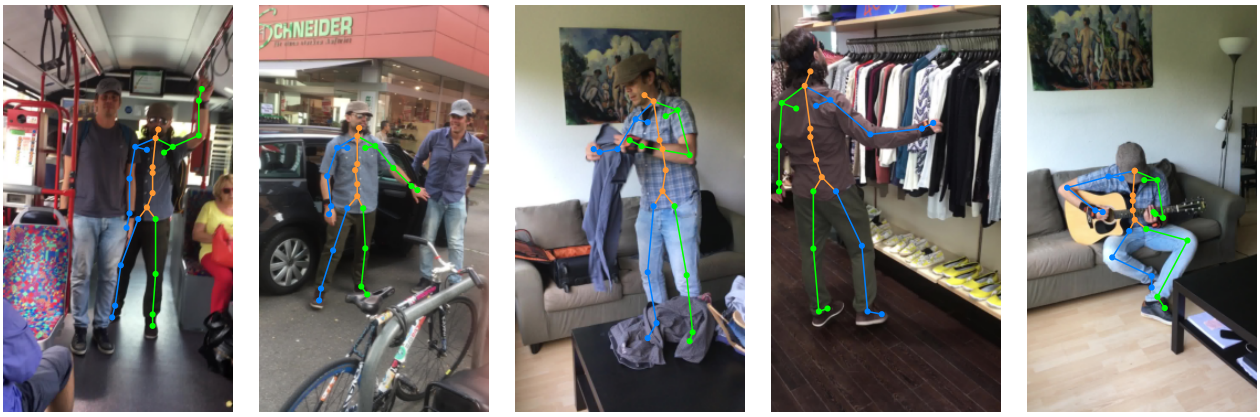

Figure 4. Pose estimation examples from our algorithm on the 3DPW dataset.

mance.

For example, in the upper red box in the figure, the COCO nose joint and the 3DPW jaw joint prototypes are closely aligned. Similarly, the lower red box contains a COCO left hip joint prototype and a 3DPW pelvis joint. As validated by domain transfer on InterHand and 3DPW, our learned embeddings effectively incorporate new skeletons without retraining of the embedding module.

3.3. Failure cases

In Fig. 6, we provide failure cases caused by unseen skeletons and poses. In (a–b), we show our COCO skeleton predictions on CrowdPose data. Since CrowdPose has a different skeletal structure than the training datasets, the predictions do not fully conform to its intended format, although the pose is reasonably well estimated in (a). Under strong occlusion (b), our method may also struggle to predict accurate skeletons, as seen in the misplacement of the estimated left foot at the location of the right foot.

Similarly, (c–d) show our AP-10K skeleton predictions on a COP3D [4] example, where the pose deviates significantly from those observed during training (e.g. the cat's head is tilted back, looking upward). In (d), our method fails to localize the cat's eyes due to this challenging, unseen pose. Incorporating temporal information could improve robustness in such cases.

References

[1] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018. 1

[2] Zigang Geng, Chunyu Wang, Yixuan Wei, Ze Liu, Houqiang Li, and Han Hu. Human pose as compositional tokens. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[3] Yanjie Li, Sen Yang, Peidong Liu, Shoukui Zhang, Yunxiao Wang, Zhicheng Wang, Wankou Yang, and Shu-Tao Xia. SimCC: A simple coordinate classification perspective for human pose estimation. In *European Conference on Computer Vision (ECCV)*, pages 89–106. Springer, 2022. 2

[4] Samarth Sinha, Roman Shapovalov, Jeremy Reizenstein, Ignacio Rocco, Natalia Neverova, Andrea Vedaldi, and David Novotny. Common Pets in 3D: Dynamic new-view synthesis of real-life deformable categories. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5

[5] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[6] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision (ECCV)*, pages 466–481, 2018. 2

[7] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose++: Vision transformer for generic body pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1

[8] Yuan Xu, Xiaoxuan Ma, Jiajun Su, Wentao Zhu, Yu Qiao, and Yizhou Wang. ScoreHypo: Probabilistic human mesh estimation with hypothesis scoring. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 979–989, 2024. 2

[9] Yuxiang Yang, Junjie Yang, Yufei Xu, Jing Zhang, Long Lan, and Dacheng Tao. APT-36K: A large-scale benchmark for animal pose estimation and tracking. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:17301–17313, 2022. 1

[10] Song-Hai Zhang, Ruilong Li, Xin Dong, Paul Rosin, Zixi Cai, Xi Han, Dingcheng Yang, Haozhi Huang, and Shi-Min Hu. Pose2Seg: Detection free human instance segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 889–898, 2019. 2

[11] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. Rethinking Semantic Segmentation: A Prototype

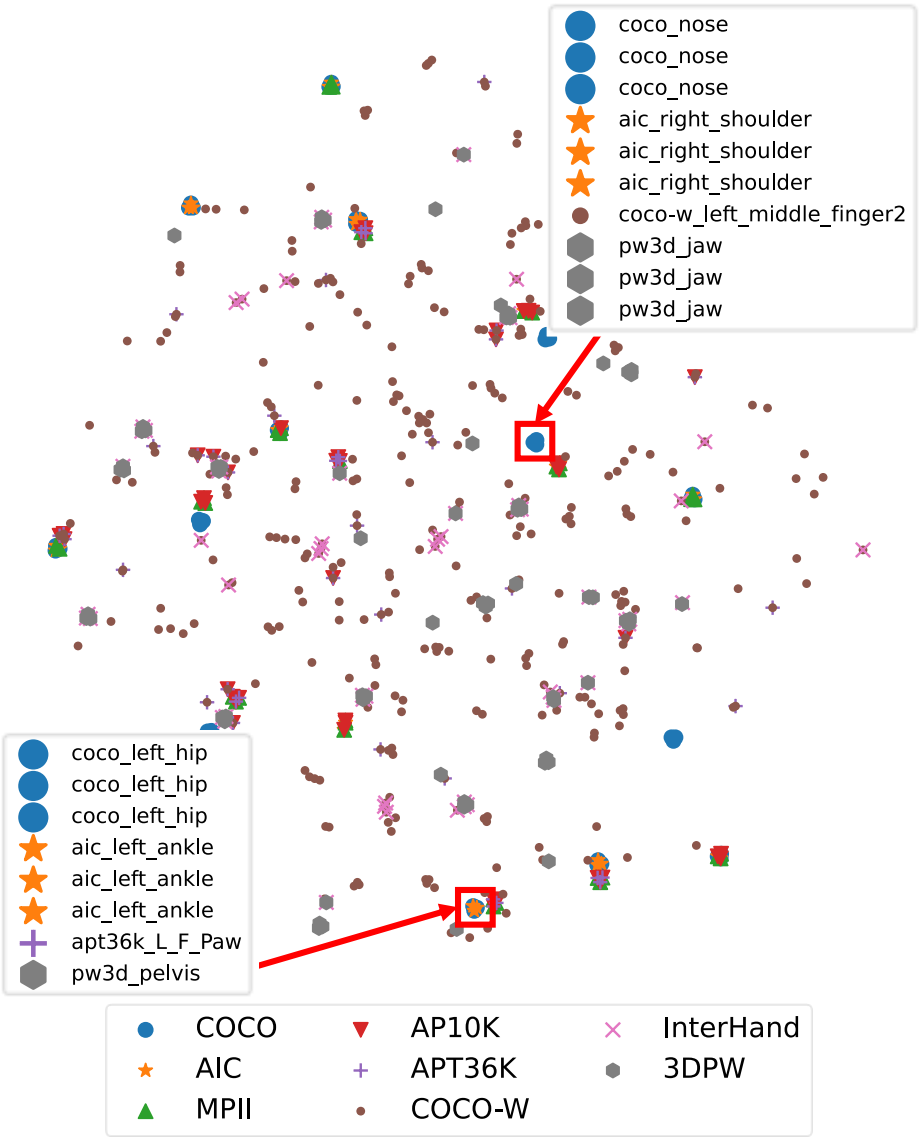

Figure 5. t-SNE visualization of the prototypes. Best viewed when zoom-in.

View. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2582–2593, 2022. 1

[12] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7571–7580, 2022. 2

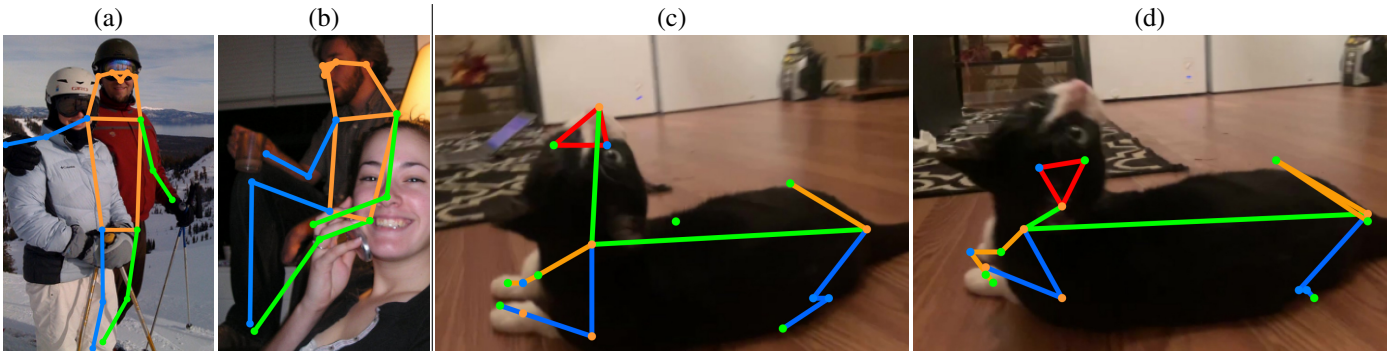

Figure 6. Example pose predictions: (a-b) CrowdPose predictions using the COCO skeleton; (c–d) COP3D predictions using the AP-10K skeleton.