DensePose 3D: Lifting Canonical Surface Maps of Articulated Objects to the Third Dimension (Supplementary Material)

Roman Shapovalov David Novotny Benjamin Graham Patrick Labatut Andrea Vedaldi
Facebook AI Research

A. Implementation details

We implemented our model using Pytorch (the code will be shared). The networks Φ in eq. (4) and Ψ (6) have the same architecture as in C3DPO [4]. In particular, they first map the input to a 1024-dimensional vector with a fully-connected layer followed by 6 residual 3-layer MLPs. The blocks have the architecture 1024–256–256–1024, each layer followed by BatchNorm and ReLU. The network parameters are optimised with SGD with momentum and weight decay. Training runs for 20 epochs on Human 3.6M, for 100 epochs on UP-3D, 200 epochs for Stanford Dogs, and 1000 epochs for the small LVIS categories, starting with the learning rate 0.003 and decreasing 10 times after 80% of epochs have passed. The momentum coefficient is 0.99 and weight decay to 0.001. The network takes sparse keypoints and is thus lean on memory compared to convolutional neural networks, which have to maintain feature maps, thus a batch size of 512 can be used with a 16 GB GPU. A forward pass for this batch size takes 0.5 seconds.

To project the predicted 3D shape back to the image plane in loss (5), we define π as orthographic projection, although the method allows using perspective projection as long as camera intrinsics are known. The input keypoints $\mathbf{Y}$ are zero-centered before passing to Φ.

The hyperparameters were set to the following values: the number of latent parts M in the segmentation model was set to 10; segmentation model initialisation parameter $\bar{\sigma} = 32$, weights of the loss functions in eq. (9) are: $w_{entropy} = 0.001$, $w_{arap} = 0.3$, $w_{canon} = 0.1$.

B. Generating keypoints for LVIS

For reconstruction of humans, we pre-process data with DensePose [1] to obtain UV coordinates defining the correspondences to the template mesh, and convert them to 2D keypoints corresponding to template mesh vertices as described in Section 3.1. For animals we instead use pre-trained CSE models [2] to obtain the per-pixel descriptors from the joint embedding space with category-specific template mesh surface. For each pixel within the object mask, we find the closest template mesh vertex in the embedding space. Let the set of pixels j that were matched to the vertex i of the template be

$$\mathcal{M}_i = \{j : i = \text{argmin}_{i'} \, \|\mathbf{e}_j - \mathbf{e}_{i'}\|\}, \tag{1}$$

where $\mathbf{e}_j$ and $\mathbf{e}_{i'}$ are the embeddings of the j-th pixel and i'-th vertex, respectively. Then, all vertices that have been matched to at least one pixel, i.e. $\mathcal{M}_i \neq \emptyset$, are considered visible. For each visible vertex, we find the corresponding 2D keypoint location as the mean coordinate of the pixels matched to it: $y_i = \mathsf{E}_{j \in \mathcal{M}_i} \, p_j$, where p_j are the coordinates in the pixel grid. This way, we ensure that all occluded surface points are marked as invisible in the DensePose 3D input.

C. Heteroscedastic reprojection loss

The annotation by CSE [2] trained on LVIS is quite noisy due to small dataset size and high amount of noise and occlusions. When fitting the model, it is generally important to use a robust loss such as L1 or Huber (we use pseudo-Huber loss in this paper wherever the norm is not explicitly specified). For CSE annotation specifically, we found important to predict the keypoint-specific variance to weigh its contribution to the loss.

To properly account for variance, we generalise L1 loss in correspondence with the maximum likelihood estimation theory similar to what Novotny *et al*. [3] did for L2 loss. First, note that L1 loss is proportional to a shifted negative log-likelihood of Laplacian distribution of the residual's L1 norm, given constant scale parameter b. In the heteroscedastic version, we do not fix those additive and multiplicative terms and consider the full NLL:

$$-\log p(y|\hat{y}, b) = \log(2b) + \frac{|y - \hat{y}|}{b}. \tag{2}$$

For numerical stability, the denominator is clipped, and the constant term is removed, which brings us to the following formulation:

$$\mathcal{L}^b_{\text{rep}}(\hat{y}, y, b) = \log b + \frac{\mathcal{L}_{\text{rep}}(\hat{y}, y)}{\max\{b, b_{\text{min}}\}}, \tag{3}$$

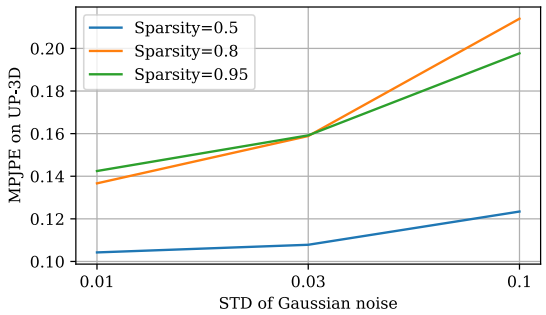

Figure 1: Reconstruction quality w.r.t. the noise for different sparsity.

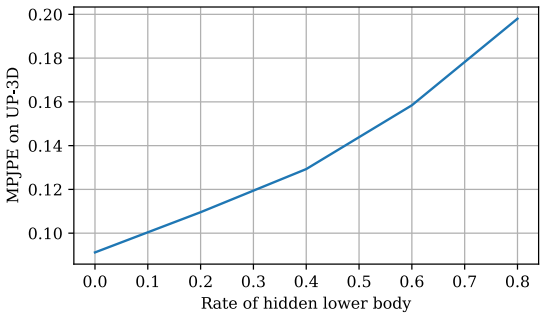

Figure 2: Reconstruction quality w.r.t. the rate of removed legs.

where b_{min} is set to 0.1, and $\mathcal{L}_{\text{rep}}(\hat{y}, y)$ is a pseudo-Huber loss $\epsilon(\sqrt{1 + (\|y - \hat{y}\|/\epsilon)^2} - 1)$, which smoothly approximates the L1 loss.

Finally, the per-keypoint uncertainty b_k is also predicted by the model based on keypoint's identity and its predicted local pose. We train a new branch that takes for each keypoint k its LBO descriptor concatenated with the predicted transformation of the corresponding template vertex. It comprises a 2-layer MLP topped with softplus. The LBO descriptor identifies the keypoint in a smooth way, while the transformation can aid uncerainty prediction because the 2D projection of the variance depends on the angle between the surface element and the camera ray.

D. Data and model for dogs

Human shape models like SMPL combine linear blendshapes with linear blend skinning; the former is responsible for modelling the body type, while the letter for articulation. While we can do the same in our model, we found that for humans it did not yield any improvement. The variation in the shape of dogs across breeds is in contrast huge, e.g. the body lenght to leg lenght ratio is different for great danes versus dachshunds. To model this shape variations, we learn a set of 5 blendshapes and apply them to the template mesh before posing it. The blendshapes are learned as a vertex-wise function of the LBO basis, in the same way as in our no-parts baseline described in Section 4.3. This parametrisation of blendshapes as a linear function of LBO descriptors again helps to achieve invariance to remeshing and avoid overfitting.

When generating the dataset, we remove cases where less than 10% of the template surface area is visible because this usually results in poor SMAL fits.

E. Robustness of training

DP3D training relies on roughly correct DensePose predictions. In case DensePose failed (e.g. due to occlusions), our model would not be able to recover. Here we investigate how the training is robust to different kinds of noise added to synthetic UP-3D. First, we simulate the case of manual annotations by adding Gaussian noise to 2D keypoints and marking some portions of them as invisible. Figure 1 shows that adding uncorrelated Gaussian noise is not harmful as long as keypoints are relatively dense, but it becomes a problem once the location of 80%+ of projected keypoints is unknown. Second, in Figure 2, we marked invisible the whole lower half of the body (in the canonical orientation) for a certain proportion of training instances. As expected, the method is less robust to occlusions than to Gaussian noise.

F. More qualitative results

See more results in comparison to baselines in Figures 3 to 6.

References

[1] Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. DensePose: Dense human pose estimation in the wild. In *Proc. CVPR*, 2018. 1

[2] Natalia Neverova, David Novotný, and Andrea Vedaldi. Continuous surface embeddings. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1

[3] David Novotný, Diane Larlus, and Andrea Vedaldi. Capturing the geometry of object categories from video supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 1

[4] David Novotny, Nikhila Ravi, Benjamin Graham, Natalia Neverova, and Andrea Vedaldi. C3DPO: Canonical 3D Pose Networks for Non-Rigid Structure From Motion. In *Int. Conf. Comput. Vis.*, 2019. 1, 4, 5

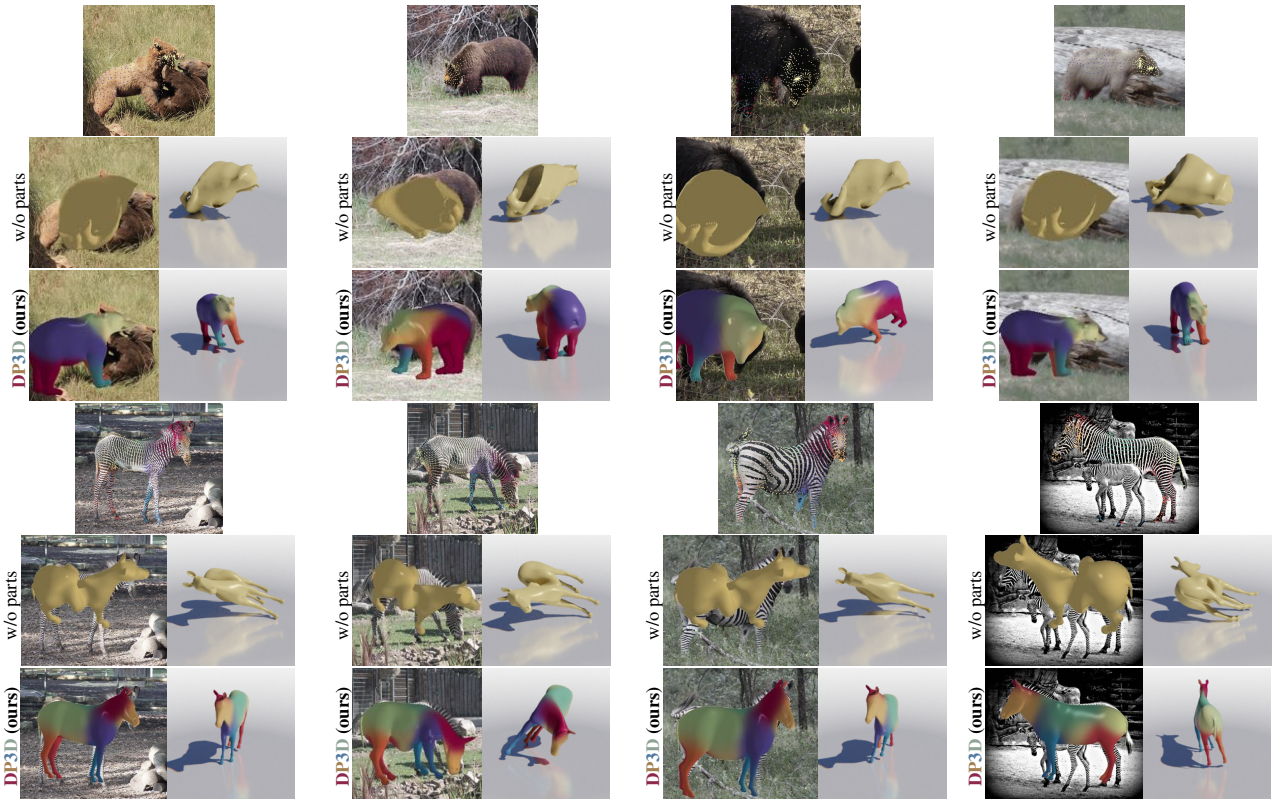

Figure 3: **Qualitative evaluation on LVIS**. Each of the rows contains, from top to bottom: input image and keypoints, the reconstruction with the linear model instead of parts segmentation, and of the proposed method.

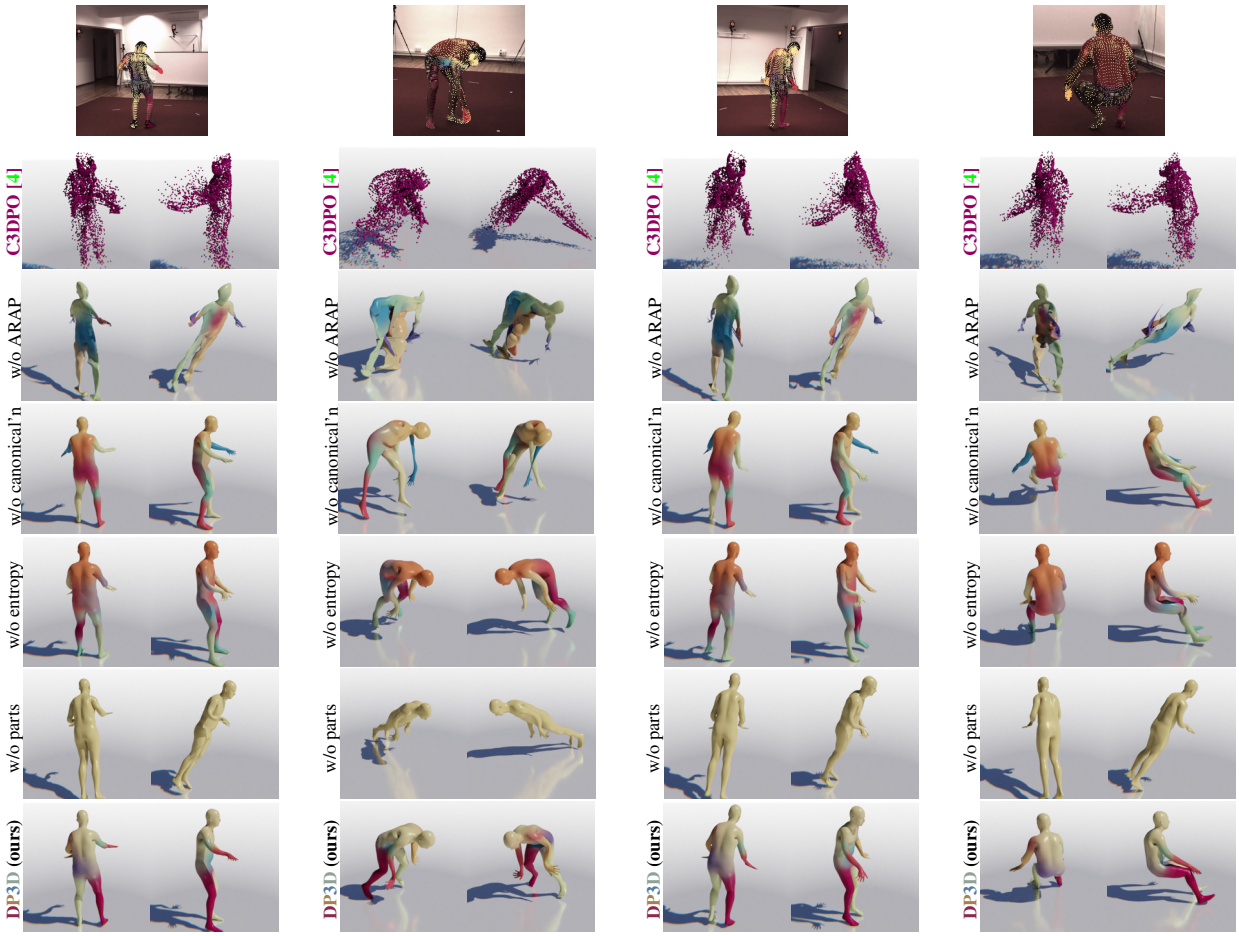

Figure 4: **Qualitative evaluation on Human 3.6M**. From top to bottom: input image and keypoints, the reconstruction of C3DPO [4], of DensePose 3D without the ARAP loss (7), without the canonicalisation loss (6), without the entropy loss (8), with the linear model instead of parts segmentation, and of the full proposed method.

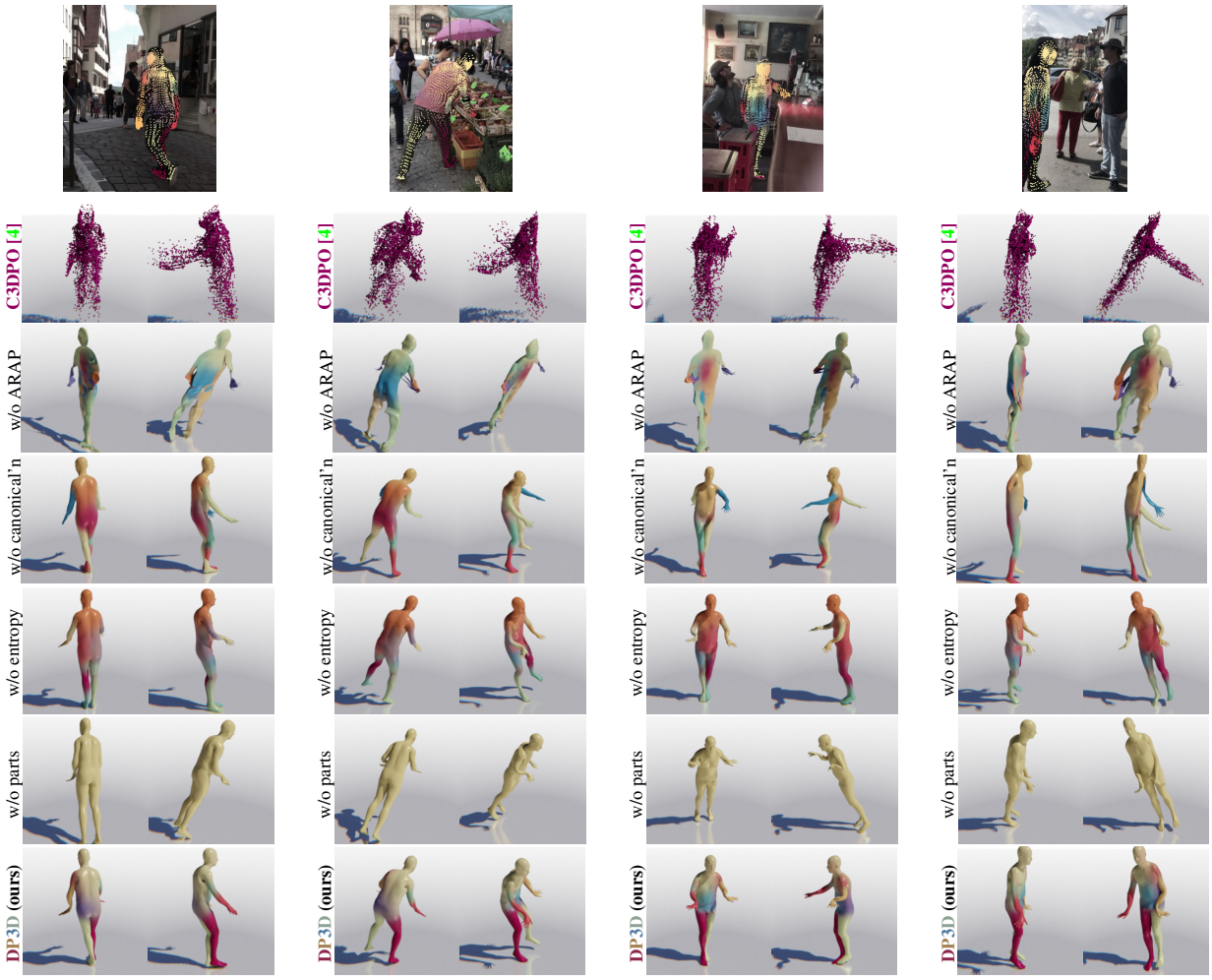

Figure 5: **Qualitative evaluation on 3DPW**. From top to bottom: input image and keypoints, the following rows show the reconstruction of C3DPO [4], the results of DP3D without corresponding losses, with the linear model instead of parts segmentation, and of the full proposed method.

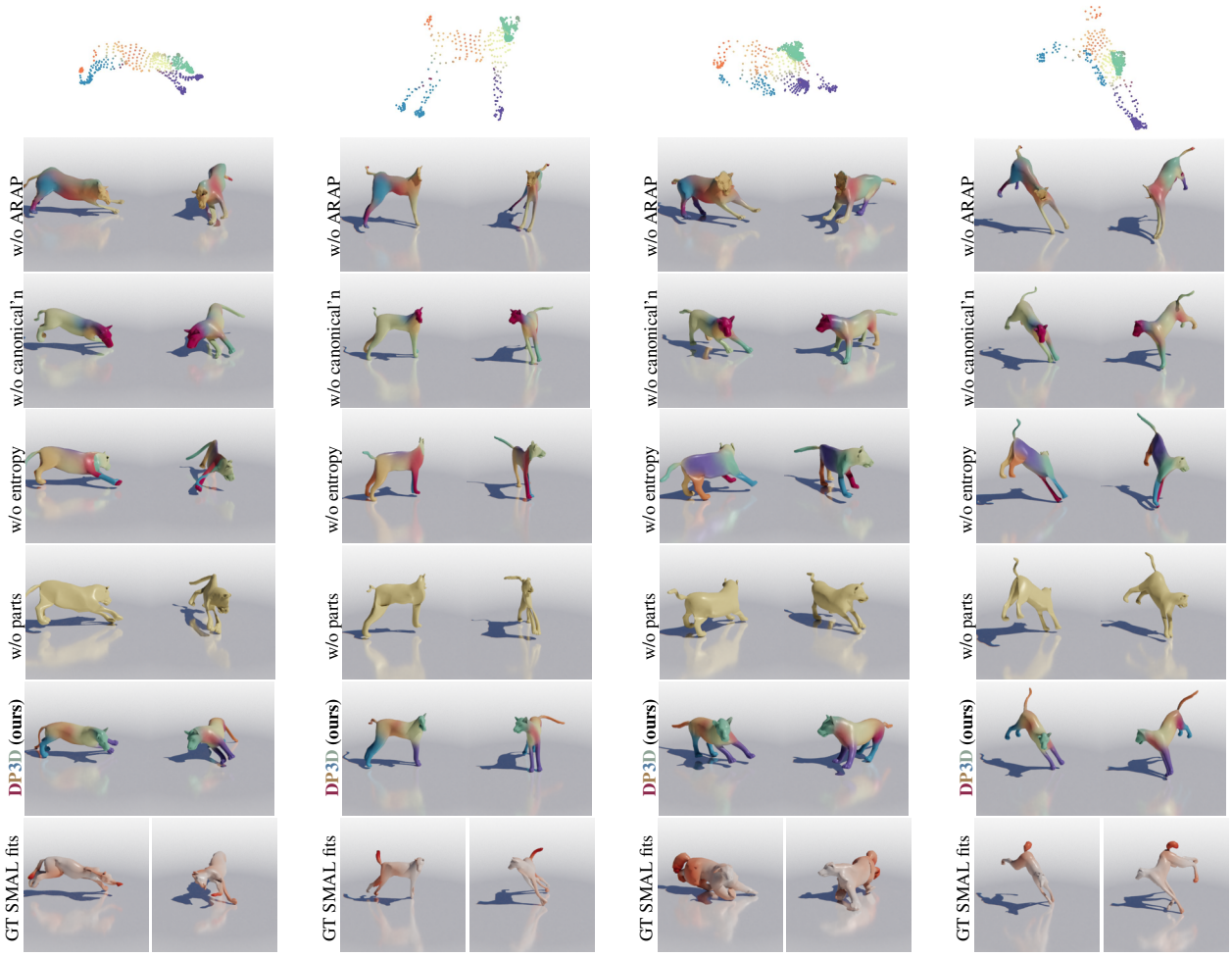

Figure 6: **Results on Stanford Dogs**. The first row shows input keypoints obtained by projecting SMPL fits showed in the last row, the following rows show the results of DP3D without corresponding losses, of the no-parts baseline and of our reconstruction from the camera's and from an alternative viewpoint, the last row color-codes errors on the "ground-truth" mesh.