Supplementary material
HandDGP: Camera-Space Hand Mesh Prediction with Differentiable Global Positioning

Eugene Valassakis*, Guillermo Garcia-Hernando

Niantic
https://nianticlabs.github.io/handdgp/

1 Implementation details

1.1 Architecture

While our method is network architecture agnostic, in our experiments we use the following. Our feature extractor consists of a ResNet50 [6] backbone pretrained on ImageNet [3], taking in an $H \times W \times C$ RGB image and outputting a $H/32 \times W/32 \times 2048$ feature map $\mathbf{F}$. Our three decoder heads then take in $\mathbf{F}$ and produce their respective outputs. While our weights decoder is described in the main paper, our 2D and 3D decoders follow the MobRecon [1] architecture. In short, for our 2D decoder, we first use a 1×1 convolution to reduce the number of features channels to one per keypoint, and then use an MLP to process each channel independently and output the (u_i, v_i) normalised pixel coordinates of each keypoint. For our 3D decoder, we first apply a 1×1 convolution to reduce the number of channels in $\mathbf{F}$, and then grid sample features from the resulting feature map using the 2D keyppoint predictions, similarly to the weights decoder. Then, we iteratively apply a set of upsampling and SpiralNet [4, 10] operations with DSConv [1], to process more and more granular features representing the mesh, all the way to the 778×3 MANO [15] vertex output.

1.2 Losses

Let $\cdot^*$ represent the ground truth version of a quantity, and N_F the number of faces in the MANO mesh. For our *relative space* losses, we follow MobRecon. As such, we apply a vertex loss $\mathcal{L}_{V^{rel}}$, a keypoint loss $\mathcal{L}_{K^{2D}}$, a norm loss $\mathcal{L}_{norm}$,

* Now at Synthesia. Work done while at Niantic.

and an edge loss $\mathcal{L}_{edge}$, such that

$$\mathcal{L}_{V^{rel}} = \frac{1}{N_V} \sum_{i}^{N_V} ||v_i^{rel} - v_i^{rel^*}||_1, \tag{1}$$

$$\mathcal{L}_{K^{2D}} = \frac{1}{N_K} \sum_{i}^{N_K} ||k_i^{2D} - k_i^{2D^*}||_1, \tag{2}$$

$$\mathcal{L}_{norm} = \frac{1}{3 \times N_F} \sum_{c \in faces} \sum_{(i,j) \subset c} \frac{v_i^{rel} - v_j^{rel}}{||v_i^{rel} - v_j^{rel}||_2} \cdot \mathbf{n}^*, \tag{3}$$

$$\mathcal{L}_{edge} = \frac{1}{3 \times N_F} \sum_{c \in faces} \sum_{(i,j) \subset c} ||v_i^{rel} - v_j^{rel}||_1 - ||v_i^{rel^*} - v_j^{rel^*}||_1, \tag{4}$$

where c represents a face in the pre-determined MANO topology and $\mathbf{n}^*$ is the normal of face c. Moreover, similarly to MobRecon [1], for each input sample we compute two views varying in scale, translation and color profile, and enforce the model's predictions are consistent between the two views. As such, we apply a 2D and 3D consistency losses,

$$\mathcal{L}_{consist2D} = \frac{1}{N_K} \sum_{i}^{N_K} ||T_{1 \to 2} k_{i,1}^{2D} - k_{i,2}^{2D}||_1, \tag{5}$$

$$\mathcal{L}_{consist3D} = \frac{1}{N_V} \sum_{i}^{N_V} ||v_{i,1}^{rel} - v_{i,2}^{rel}||_1, \tag{6}$$

where $T_{1 \to 2}$ is an affine transformation mapping between the two views. We note that, unlike MobRecon [1], our two views share the same rotation, so we do not need to apply a rotation corrective transformation in $\mathcal{L}_{consist3D}$.

For our *camera space* losses, we have a root translation loss $\mathcal{L}_t$, a keypoint consistency loss $\mathcal{L}_{Kconsist}$ and a 2D vertex loss $\mathcal{L}_{V^{2D}}$ such that,

$$\mathcal{L}_t = ||t - t^*||_1, \tag{7}$$

$$\mathcal{L}_{V^{2D}} = \frac{1}{N_V} \sum_{i}^{N_V} ||\Pi(v_i^{cs}) - \Pi(v_i^{cs^*})||_1 \tag{8}$$

$$\mathcal{L}_{Kconsist} = \frac{1}{N_K} \sum_{i}^{N_K} ||\Pi(k_i^{cs}) - \Pi(k_i^{2D})||_1, \tag{9}$$

where $\Pi(\cdot)$ represents the perspective projection operation using ground truth intrinsics, and k_i^{cs} is obtained from v_i^{cs} by applying $\mathbf{J}_{reg}$. Our final loss $\mathcal{L}$

can then be written as:

$$\begin{aligned}\mathcal{L} = &1.0 \cdot \mathcal{L}_{V^{rel}} + 1.0 \cdot \mathcal{L}_{K^{2D}} + 0.05 \cdot L_{norm} \\ &+ 0.5 \cdot L_{edge} + 1.0 \cdot \mathcal{L}_{consist2D} + 1.0 \cdot \mathcal{L}_{consist3D} \\ &+ 1.0 \cdot \mathcal{L}_t + 1.0 \cdot \mathcal{L}_{V^{2D}} + 0.5 \cdot \mathcal{L}_{Kconsist},\end{aligned} \tag{10}$$

where the weights for each loss have been empirically determined.

1.3 Preprocessing.

As a preprocessing step for all our models, we crop the original image around the hand and then resize the resulting crop to 224×224. For all our rectification-based experiments, we use a canonical focal length of 1000, to which we map all our images. To train our models, we initially train with the *relative space* losses until convergence and then fine-tune with all losses as presented in Equation 10. The reason for this curriculum is that our 2D-3D correspondence-based algorithm relies on the rigid-body assumption. At the beginning of training, this assumption is too strongly violated, leading to unstable training.

1.4 Augmentations

. During training, we augment the images by applying a random shift, scale, and blur, as well as random changes in brightness, contrast, saturation, and hue to each contrastive sample (cf. Equations 5 and 6). We also apply a random rotation augmentation, which remains consistent between both contrastive samples.

2 Comment on $\mathbf{A}^T\mathbf{A}$ invertibility.

In general, for a $m \times n$ matrix $\mathbf{A}$, $\mathbf{A}^T\mathbf{A}$ is invertible if $m \geq n$ and $\mathbf{A}$ is full rank, that is with linearly independent columns. The first condition holds ($m = 2 \times N_k = 42, n = 3$). However, the second condition is violated if and only if all the 2D projected keypoints share the same exact coordinates. Therefore, in theory, we cannot ensure that $\mathbf{A}^T\mathbf{A}$ is always invertible. In practice, to avoid singularities that could make training unstable, we first pretrain without the DGP module, reducing the probability of this happening. In the rare case that we still face a singularity, we can reject the gradients from that batch.

3 Comment on extending our method to use bone length

. Providing actual anthropometric measurements of the input hand can reduce depth-scale ambiguity [7, 16, 17] by removing a source of uncertainty. However, this information is usually unavailable at test time, limiting its application. One way to extend our work is by providing a bone reference length l_{ref} and scaling

V^{rel} and K^{3D} with a factor $s = \frac{l_{\text{ref}}}{l_{\text{pred}}}$, where l_{pred} is the predicted bone length using K^{3D}. The 2D–3D correspondences used to derive the DGP t^* solution in Equations (2–7) then use the scaled correspondences $\mathcal{M}' = \left\{\left(k_i'^{\text{3D}}, k_i^{\text{2D}}\right)\right\}_{i=1}^{N_K}$, with $k_i'^{\text{3D}} = s \cdot k_i^{\text{3D}} = s \cdot [x_i, y_i, z_i]^{\text{T}}$ instead of $\mathcal{M}$.

4 Image Rectification Visualization

. In Figure 1 we compare the rectified input image to the original. Black borders may appear due to image minification (see the 2nd and 4th examples). The amount of black border depends on camera parameters and hand position.

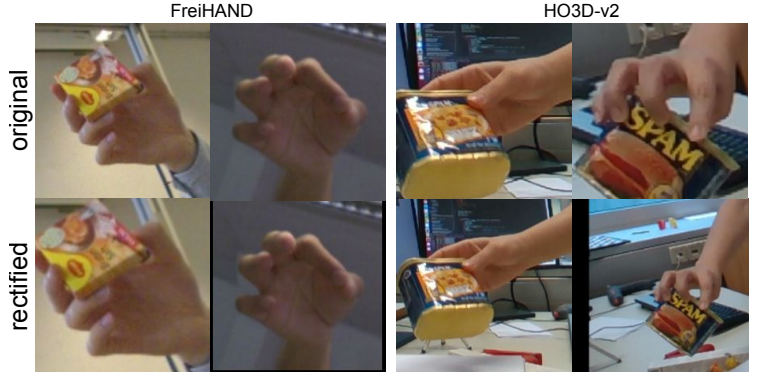

Fig. 1: Network input images with and without rectification.

5 Evaluating Generalizability: Applying our Framework to a Different Root-Relative Method.

In the main paper, we built our framework upon the MobRecon [12] root-relative mesh predictor. In this experiment, we aim to evaluate whether our learnings can be applied to a different method. For this purpose, we select the CMR model from [2], which predicts 3D relative meshes, UV maps for 2D keypoints, and hand silhouettes, and adapt it to our framework. In the original CMR the hand root is predicted using correspondences and hand silhouettes within a post-processing optimization function.

In this experiment add our Image Rectification module and replace the optimization function with our DGP model, then train the entire model end-to-end. We utilize the code provided by the authors and adapt it to our framework.

Method	Image Rectification	End-to-End Training	Root-Relative RS-MJE↓	Root-Relative RS-MVE↓	Camera-Space CS-MJE↓	Camera-Space CS-MVE↓
lighterblue CMR [2]	✗	✗	7.4	7.5	54.1	54.1
lighterblue CMR [2] + Rect.	✓	✗	7.6	7.6	51.2	51.2
lighterblue CMR [2] + Rect. + DGP	✓	✓	7.6	7.7	50.2	50.2
lightblue MobRecon [12] +Rect.+DGP	✓	✓	7.4	7.6	46.3	46.3

Table 1: Using a different root-relative backbone for our method on FreiHAND dataset [18]. The *'Image Rectification'* column indicates whether the training images are rectified with our proposed approach. *'End-to-end Training'* denotes whether gradients flow through the global positioning function during training. The task we care about for 3D interactions is quality in the camera space where we outperform all the baselines and variants, validating our approach.

To learn the weights of DGP, we introduce another branch with a weight decoder from the latent features ($512 \times 7 \times 7$) that feed into the UV branch (the final decoder of their three backbones). We use a ResNet18 backbone with their SG architecture, leaving the rest of the parameters unchanged and training the models for 25 epochs. We use the same cropping / augmentation technique than in our main paper baseline and full method and add the corresponding losses in addition to those from the original CMR.

The ablation results on FreiHAND dataset are presented in Table 1. We observe that CMR benefits from both our rectification step and DGP module. Interestingly, we observe only a minor degradation in relative results, which could be due to the use of semantic hand prediction that emphasizes learning features from the silhouette. Our CMR results are slightly inferior to those reported in the main paper, suggesting that our cropping/augmentation might not be as effective as theirs, as this was the only significant change from the original codebase. This degradation could potentially impact our full method, indicating that our results might improve by modifying our image augmentation approach.

In conclusion, our results confirm that our framework is agnostic to the root-relative approach, and other methods could potentially benefit from our findings.

6 State of the Art Comparison: Qualitative Results

In this section, we present qualitative results on HO3D-v2 test set that complements the Table 4 of the main paper. Here we show results for our method, Hasson *et al*. [21], which showed the lowest errors among competing state-of-the-art approaches as depicted in Table 4 (main paper), and MobRecon [12], which serves as our main baseline since we build upon their relative root-prediction model.

Note that we do not have access to ground-truth values to visualize ground-truth meshes, as the evaluation is carried out on a public server. However, the authors provide ground-truth hand root (wrist) which we use for 3D error visualization. These ground-truth labels are often in the literature use to position the hand in 3D, which leads to lower error values. In all our results, we use predicted values instead.

The qualitative results are shown in Figures 2, 3, 4 and 5. We observe that our method generally outperforms previous approaches in both 2D projections and camera-space 3D placement. Visualizing the errors in 3D presents a different picture compared to just 2D projections, which is often the focus of evaluation in the literature. While we achieve better results than the current state of the art, the problem is far from being solved. We also present negative results in Figure 6, where all three methods fail, illustrating the difficulty of the problem.

Video examples are available in the supplementary video.

Fig. 2: Camera-Space Mesh Prediction Qualitative Result Comparison. **Top row:** 2D hand mesh projection visualization on a test image. **Middle and bottom rows:** 3D visualization of meshes from the top row image, taken from different viewpoints for visualization purposes. The red sphere represents the ground-truth hand root (wrist) value, while the spheres in different colors represent the predicted root values by the respective methods. **Middle row:** HandDGP vs. Hasson *et al.* [21]. **Bottom row:** HandDGP vs. MobRecon [12]. We observe that our method provides both better 2D hand projections and more accurate 3D camera-space positions of the hand root compared to the other two methods.

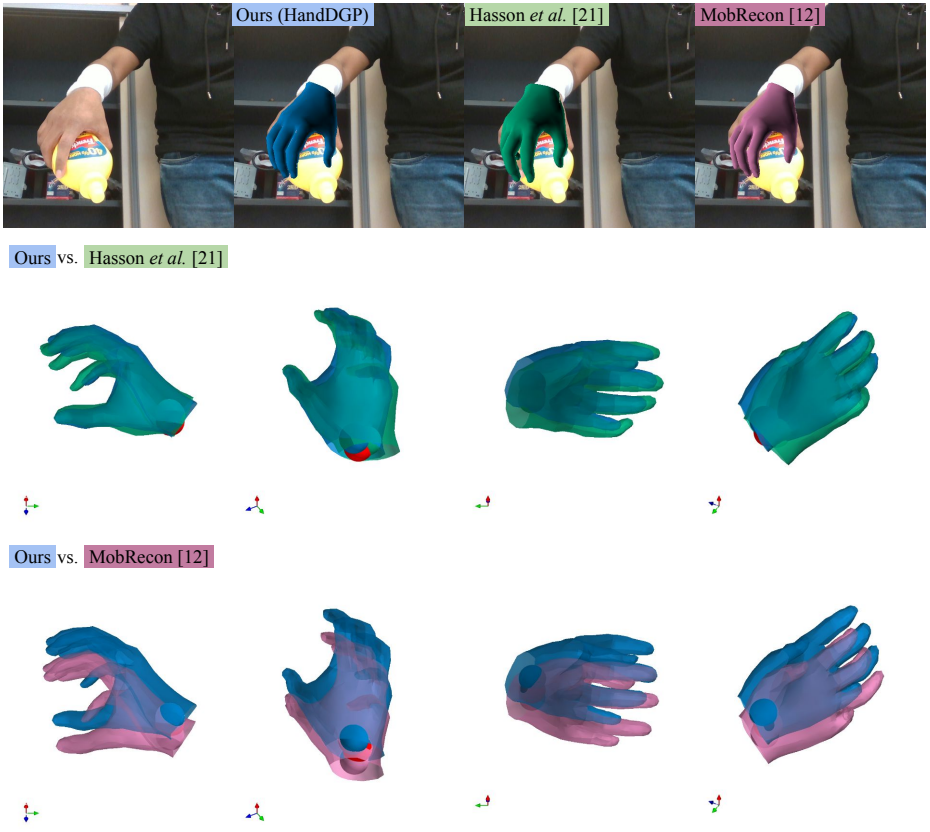

Fig. 3: Camera-Space Mesh Prediction Qualitative Result Comparison. **Top row:** 2D hand mesh projection visualization on a test image. **Middle and bottom rows:** 3D visualization of meshes from the top row image, taken from different viewpoints for visualization purposes. The red sphere represents the ground-truth hand root (wrist) value, while the spheres in different colors represent the predicted root values by the respective methods.**Middle row:** HandDGP vs. Hasson *et al.* [21]. **Bottom row:** HandDGP vs. MobRecon [12]. We observe similar accuracy across methods.

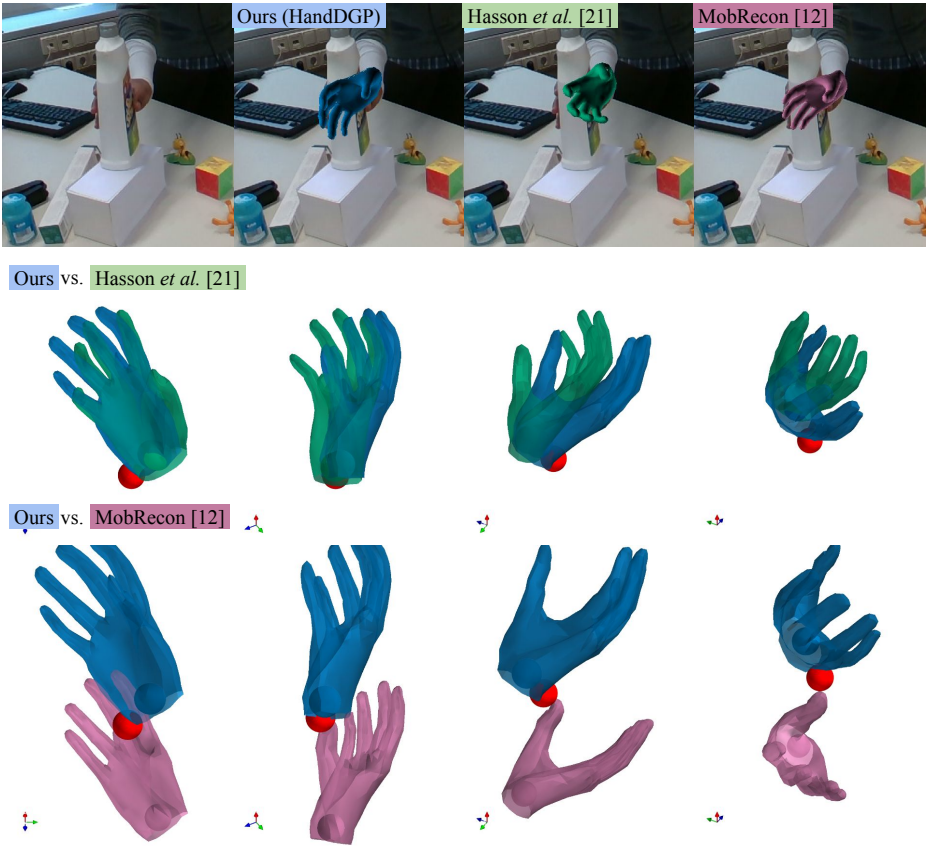

Fig. 4: Camera-Space Mesh Prediction Qualitative Result Comparison. **Top row:** 2D hand mesh projection visualization on a test image. **Middle and bottom rows:** 3D visualization of meshes from the top row image, taken from different viewpoints for visualization purposes. The red sphere represents the ground-truth hand root (wrist) value, while the spheres in different colors represent the predicted root values by the respective methods. **Middle row:** HandDGP vs. Hasson *et al.* [21]. **Bottom row:** HandDGP vs. MobRecon [12]. In this image, our method outperforms the other two methods in both 2D projections and 3D hand placement. We observe that the 3D prediction from MobRecon [12] is significantly incorrect.

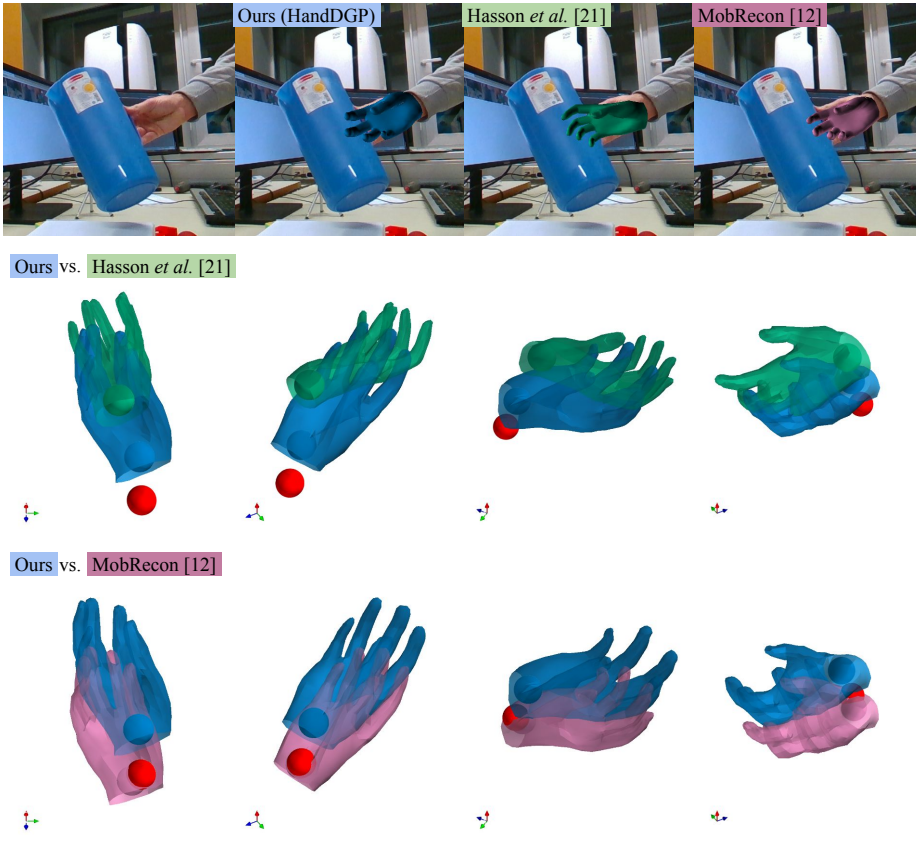

Fig. 5: Camera-Space Mesh Prediction Qualitative Result Comparison. **Top row:** 2D hand mesh projection visualization on a test image. **Middle and bottom rows:** 3D visualization of meshes from the top row image, taken from different viewpoints for visualization purposes. The red sphere represents the ground-truth hand root (wrist) value, while the spheres in different colors represent the predicted root values by the respective methods. **Middle row:** HandDGP vs. Hasson *et al*. [21]. **Bottom row:** HandDGP vs. MobRecon [12]. In this image, our method demonstrates better 3D root prediction than Hasson *et al*. [21]; however, MobRecon [12] performs even better. We observe that the predicted thumb finger from Hasson *et al*. [21] appears more accurate, likely because it has been trained with object models and contact losses.

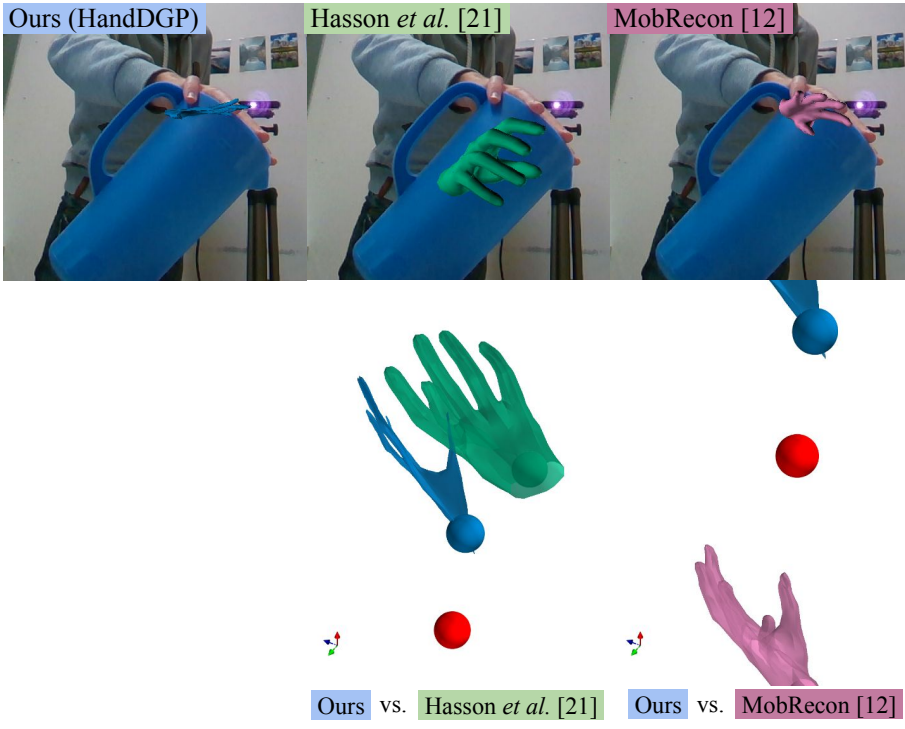

Fig. 6: Camera-Space Mesh Prediction Qualitative Result Comparison: Method failure. **Top row:** 2D hand mesh projection visualization on a test image. **Middle rows:** 3D visualization of meshes from the top row image, taken from different viewpoints for visualization purposes. The red sphere represents the ground-truth hand root (wrist) value, while the spheres in different colors represent the predicted root values by the respective methods. We observe that all the methods fail to predict reasonable hand meshes.

7 More Experiments Details

Fig. 7: HO3Dv-2: Camera-space hand mesh qualitative results on validation images. We observe how our learned keypoints weights that ponder the correspondences in the DGP module tend to be lower (darker) on occluded hand joints.

7.1 HO3D-v2

We report additional camera-space ablation results on an different dataset, HO3D v2 [5]. In this dataset, the test set is not publicly available, and the evaluation is conducted on a public server. The server provides results in camera-space,

root-relative, and aligned formats. However, participants are given ground truth camera-space hand translation values, and previous work typically reports results using this ground truth. We have no way of knowing which participants on the leaderboard have used this ground truth. In this experiment, different to previous work, *e.g.* [1, 7, 12], we use our method to predict the hand translation and submit the results to the server using predicted values, not ground truth.

We utilize version 2 of the HO3D dataset [5]. This dataset comprises real images capturing 3D hand-object interactions, with 66,034 images in the training set and 11,524 in the test set with MANO [15] model hand mesh annotations. Hands in this dataset suffer of severe occlusions caused by the manipulated object. Evaluations are conducted on the official server. Only ground truth root values are provided for the test set, and we use them to calculate translation and depth errors. Finally, for this experiment we change the canonical focal length to $f^c = 500$.

We present additional qualitative results in Figure 7. We observe that keypoints occluded by objects are typically assigned lower weights by the network, as they are considered less reliable.

7.2 Human3.6M

To further assess the generalizability of our method, we adapt our implementation to predict full-body meshes using the SMPL parametric model annotations [13]. The only change we make to our method is to appropriately adjust $\mathbf{J}_{\text{reg}}$ and the keypoint and vertex heads to predict 29 2D keypoints and 6980 vertex meshes respectively. We keep all other hyperparameters constant, except for the canonical focal length, $f^c = 400$, and the image augmentations, which are adjusted to center on the bodies.

For this experiment, we utilize the Human3.6M dataset [8], a 3D body pose benchmark consisting of 3.6 million frames with 3D body joint coordinates, and SMPL annotations provided by [11]. We use S1, S5, S6, S7, and S8 for training, while S9 and S11 are used for testing. The training set is subsampled by a factor of 5, resulting in a final number of 309,309 images for training and 2,142 for testing. Note that for this dataset, previous work [2, 9, 11, 13, 14] typically only reports root-aligned and Procustres metrics, in contrast we report camera-space metrics.

We present additional qualitative results in Figure 8. Similar conclusions to those drawn in previous experiments can be observed. We observe that our method can also be used in the problem of camera-space full body mesh prediction. Interestingly, we note that the method tends to assign higher weights to the keypoints in the head, feet, and torso-pelvis, which are likely to be less occluded or variable compared to, for example, the arms and hands.

Fig. 8: Human3.6M: Camera-space full body mesh qualitative results on test iamges. We observe that our method can also be used in the problem of camera-space full body mesh prediction. Interestingly, we note that the method tends to assign higher weights to the keypoints in the head, feet, and torso-pelvis, which are likely to be less occluded or variable compared to, for example, the arms and hand.

8 Source Code

Code repository and trained models can be found in https://github.com/nianticlabs/HandDGP.

References

1. Chen, X., Liu, Y., Dong, Y., Zhang, X., Ma, C., Xiong, Y., Zhang, Y., Guo, X.: Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image. In: CVPR (2022) 1, 2, 13
2. Chen, X., Liu, Y., Ma, C., Chang, J., Wang, H., Chen, T., Guo, X., Wan, P., Zheng, W.: Camera-space hand mesh recovery via semantic aggregation and adaptive 2D-1D registration. In: CVPR (2021) 13
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009) 1
4. Gong, S., Chen, L., Bronstein, M., Zafeiriou, S.: Spiralnet++: A fast and highly efficient mesh convolution operator. In: ICCVW (2019) 1
5. Hampali, S., Rad, M., Oberweger, M., Lepetit, V.: Honnotate: A method for 3D annotation of hand and object poses. In: CVPR (2020) 12, 13
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 1
7. Huang, L., Lin, C.C., Lin, K., Liang, L., Wang, L., Yuan, J., Liu, Z.: Neural voting field for camera-space 3D hand pose estimation. In: CVPR (2023) 3, 13
8. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. TPAMI (2013) 13
9. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: CVPR (2018) 13
10. Lim, I., Dielen, A., Campen, M., Kobbelt, L.: A simple approach to intrinsic correspondence learning on unstructured 3d meshes. In: ECCV (2018) 1
11. Moon, G., Lee, K.M.: I2L-MeshNet: Image-to-lixel prediction network for accurate 3D human pose and mesh estimation from a single RGB image. In: ECCV (2020) 13
12. Park, J., Oh, Y., Moon, G., Choi, H., Lee, K.M.: Handoccnet: Occlusion-robust 3d hand mesh estimation network. In: CVPR (2022) 13
13. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3D hands, face, and body from a single image. In: CVPR (2019) 13
14. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Coarse-to-fine volumetric prediction for single-image 3d human pose. In: CVPR (2017) 13
15. Romero, J., Tzionas, D., Black, M.J.: Embodied hands: Modeling and capturing hands and bodies together. ACM TOG (2017) 1, 13
16. Spurr, A., Iqbal, U., Molchanov, P., Hilliges, O., Kautz, J.: Weakly supervised 3d hand pose estimation via biomechanical constraints. In: ECCV (2020) 3
17. Zhou, Y., Habermann, M., Xu, W., Habibie, I., Theobalt, C., Xu, F.: Monocular real-time hand shape and motion capture using multi-modal data. In: CVPR (2020) 3
18. Zimmermann, C., Ceylan, D., Yang, J., Russell, B., Argus, M., Brox, T.: FreiHAND: A dataset for markerless capture of hand pose and shape from single RGB images. In: ICCV (2019) 5