HaMuCo: Hand Pose Estimation via Multiview Collaborative Self-Supervised Learning Supplementary Materials

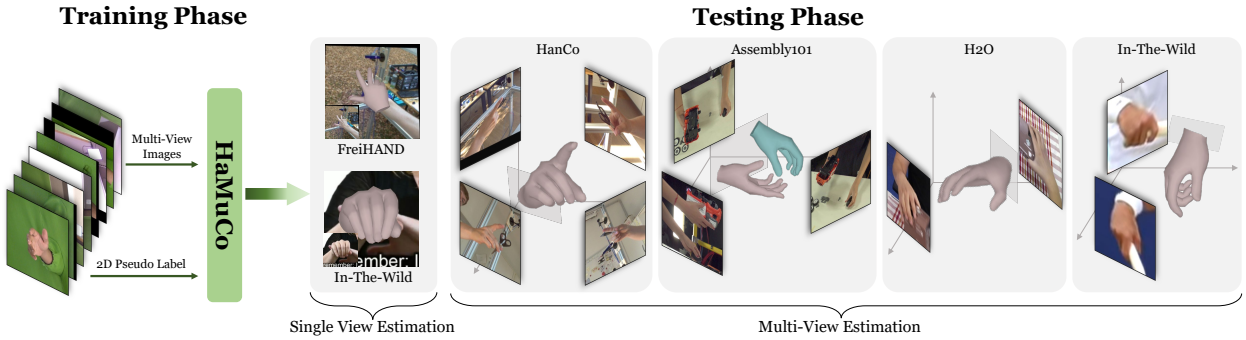

Figure A. Our method takes multi-view images with 2D pseudo labels for training. From the results on public datasets [10, 15, 22, 23] and in-the-wild images, we demonstrate that our method can estimate accurate 3D hand pose with single- or arbitrary multi-view images.

In the supplemental material, we provide:

§A Video Demo.

§B Implementation Details.

§C More Experiments and Results.

§D Discussions.

A. Video Demo

We provide additional sequential qualitative results in the attached video.

B. Implementation Details

B.1. Single-View Network

As described in our paper, we only adopt a simple single-view estimation network for our framework. The details of our single-view network are shown in Fig. B. The network only consists of a backbone (ResNet [5]) for image feature extraction, a regression head for regressing the MANO [14] parameters, and a MANO layer for parameters decoding to obtain hand mesh. Besides, the regression head is quite simple, only stacking 1 global average pooling (GAP) layer, 2 fully-connected layers, and 1 Leaky-ReLU layer.

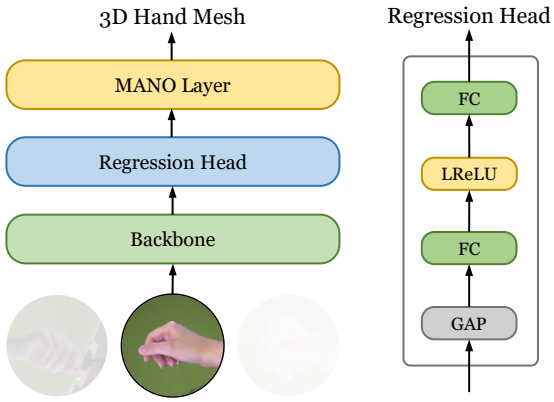

Figure B. The details of our single-view estimation network.

B.2. Multi-View Graph Feature Extraction Module

Here, we will provide more details about our multi-view graph feature extraction module. The multi-view graph extraction conducts view-shared graph extraction (VSGFE) for each view at first. VSGFE consists of three view-shared modules, a location embedding (LE) module, a spatial-aware initial graph building (SAIGB) module [21], and a joint feature sampler (JFS). LE uses an MLP to map the predicted 3D joints $\boldsymbol{P}_i \in \mathbb{R}^{21\times 3}$ and MANO pose parameters (without root joint) $\boldsymbol{\theta}_i^{'} \in \mathbb{R}^{15\times 3}$ from the single-view estimation network to the joints embeddings $\boldsymbol{G}_i^1 \in \mathbb{R}^{21\times 64}$. SAIGB first uses an MLP to scale the channel number of the high-level feature maps $\boldsymbol{H}_i^4 \in \mathbb{R}^{2048\times 8\times 8}$ to a dimension 21×8. Then, it reshapes the features to obtain $\boldsymbol{G}_i^2 \in \mathbb{R}^{21\times 512}$. Motivated by [19, 20], we design a joint feature sampler (JFS) to sample the joint-aligned features

from the middle-level feature maps. The details of our JFS are shown in Fig. C. Given the 3D coordinates of hand joints, we calculate its 2D projections on the feature map using weak perspective projection, then gather the features from nearby pixels via bilinear interpolation. In particular, we sample the joint-aligned features from three levels of the feature maps $\{\boldsymbol{H}_i^j\}_{j=1}^3$ to obtain $\boldsymbol{G}_i^3 \in \mathbb{R}^{21\times1792}$. After concatenation and stack, we obtain multi-view graph feature $\boldsymbol{G} \in \mathbb{R}^{21\times2368}$.

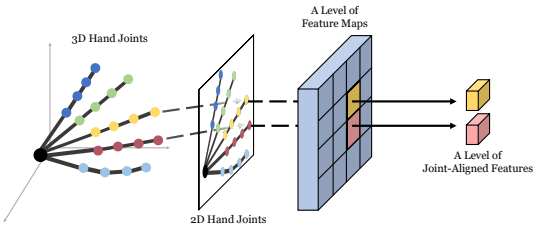

Figure C. Illustration of our joint feature sampler (JFS) sampling a level of the joint-aligned features for 2 joints.

B.3. Architecture Details

Tab. A shows the details of our complete architecture. Unless otherwise specified, MLP denotes using 2 fully-connected layers and 1 Leaky-ReLU layer (same as the regression head in Fig. B without GAP). We use 2 layers of CVA and VSF in the dual-branch cross-view interaction module (e.g. CVA-1 denotes the first CVA branch).

B.4. Loss Weights

To balance multiple loss functions, we introduce α and γ in our loss function. For all of our experiments, we set $\alpha = 0.01$ and $\gamma = 100$. It is worth mentioning that adjusting α to a correct scale is important for self-supervised learning because α balances the strength of hand-prior information provided by the MANO and the trustworthiness of pseudo labels. When the pseudo labels are reliable, we can reduce α to trust the pseudo labels more. Otherwise, we should enlarge α to use MANO to regularize irrational poses.

B.5. Hand Center Coordinate System

As shown in Fig. A, our method can be used for multi-view inference with or without camera extrinsics. If the camera extrinsics are known (HanCo [22] and Assembly101 [15]), the coordinate system of the hand center is the world coordinate system. If the extrinsics are not available (H2O [10] and in-the-wild), we choose one view as the reference view, and the center is located in this reference view coordinate system.

#Out	#In	Shape	Operation	Notation
Backbone:				
1	/	(8, 3, 256, 256)	Input	$\boldsymbol{I}$
2	1	(8, 64, 64, 64)	ResLayer	
3	2	(8, 256, 64, 64)	ResBlock1	$\boldsymbol{H^1}$
4	3	(8, 512, 32, 32)	ResBlock2	$\boldsymbol{H^2}$
5	4	(8, 1024, 16, 16)	ResBlock3	$\boldsymbol{H^3}$
6	5	(8, 2048, 8, 8)	ResBlock4	$\boldsymbol{H^4}$
Single-View Decoder:				
7	6	(8, 2048)	GAP	
8	7	(8, 48)	MLP	$\boldsymbol{\theta}$
9	7	(8, 10)	MLP	$\boldsymbol{\beta}$
10	7	(8, 3)	MLP	$s, \boldsymbol{t}$
11	8,9	(8, 778, 3)	MANO	$\boldsymbol{M}$
12	11	(8, 21, 3)	Regressor	$\boldsymbol{P}$
Multi-View Graph Feature Extraction:				
13	8,12	(8, 21, 64)	LE	$\boldsymbol{G}^1$
14	6	(8, 21, 512)	SAIGB	$\boldsymbol{G}^2$
15	3,4,5	(8, 21, 1792)	JFS	$\boldsymbol{G}^3$
16	13,14,15	(8, 21, 2368)	Concat	
17	16	(168, 2368)	Reshape	$\boldsymbol{G}$
Dual-Branch Cross-View Interaction:				
18	17	(168, 2368)	CVA-1	
19	17	(168, 2368)	VSF-1	
20	17,18,19	(168, 2368)	Add	
21	20	(168, 2368)	CVA-2	$F_t(\boldsymbol{G})$
22	20	(168, 2368)	VSF-2	$\boldsymbol{C}'$
23	20,21,22	(168, 2368)	Add	$\boldsymbol{G}^*$
Parameters Regression:				
24	23	(168, 32)	MLP	
25	24	(8, 672)	Reshape	
26	25	(8, 48)	MLP	$\boldsymbol{\theta}^*$
27	25	(8, 3)	MLP	$s^*, \boldsymbol{t}^*$
28	9,26	(8, 778, 3)	MANO	$\boldsymbol{M}^*$
29	28	(8, 21, 3)	Regressor	$\boldsymbol{P}^*$

Table A. The architecture of our whole network. We show the output shapes after every operation when adopting ResNet-50 as the backbone and taking 8 views of images of resolution 256×256 as the input. #Out and #In denotes the output and input index of this operation. In the last column, we specify those outputs that have notations in our paper.

C. Experiments and Results

C.1. Different Settings

We show the different assumptions of our experiments in Tab. B. There are generally two settings, and in both settings, we do not require GT centers. For single-view inference, which corresponds to Tab.1 and Tab.2 in the main text.

Scheme	Stage	Intrinsic	Extrinsic	GT Center
1	Train	×	×/✓	×
	Test	×	×	×
2	Train	×	✓	×
	Test	✓	✓	×

Table B. Different assumptions for HaMuCo.

Extrinsics are optionally used during the training phase, and all experiments that utilize camera extrinsics are marked with 📷. The multi-view inference is an additional benefit of our method, corresponding to Tab.3. Only in the test phase, do we require both intrinsic and extrinsic to obtain the 3D pose of absolute scale.

C.2. Datasets

Assembly101 [15] is an action recognition dataset that consists of 4,321 videos recording different persons manipulating toys. It is recorded by 8 simultaneous static cameras and 4 egocentric cameras. We only use 8 sequences of 8 static cameras for training and present the qualitative results on an additional sequence.

H2O [10] provides synchronized multi-view RGB-D images with two hands manipulating objects. The data captured by 4 static cameras and 1 egocentric camera consists of 344,645 frames for training, 73,380 frames for validation and 153,620 frames for testing. We only evaluate our cross-dataset performance on this dataset using one sequence with 1 egocentric camera and 2 static cameras.

C.3. Pseudo Labelling

We obtain the 2D joints pseudo labels at an offline stage through an implementation[1] of OpenPose [1, 16]. For HanCo [22], we directly input the images with the original size due to the images having been cropped already. For Assembly101 [15], we use a hand detector to locate and crop the hands. Then, we input the cropped images to obtain the pseudo labels.

C.4. Model Analysis

Different view number for training and inference. Here, we explain the camera settings of the experiments evaluating the performance of our models using different view numbers for training and inference (Fig. 3 in the main submission). Specifically, all the camera settings follow two rules. First, we only test the performance on a specific view for fair comparisons, considering only one specific view is available for all the experimental settings. Second, we choose camera combinations that cover a wider field of vision so that more information can be provided when the camera number has been determined.

Multi-view weakly-supervised learning. Our method can also be applied to weakly-supervised learning. Therefore, we conduct an experiment to show the performance of our model using weak 2D supervision. Considering the 2D labels from different views of the HanCo dataset are projected by the same 3D label, using all the 2D labels as weak supervisions may introduce implicit 3D supervision. Therefore, we only utilize the 2D labels from a specific view for weakly-supervised learning. During the training, we set the confidence of the labels to 1. As shown in Tab. C, when incorporating the label of a view, the performance can be improved. The performance improvement of single-view and interaction without alignments is not significant compared to others. The reason may be two folds. First, it is difficult to obtain a correct rotation from single-view inference. Second, multi-view inference without extrinsics is not able to well correct the global rotation error from every single view. In summary, our method can benefit from available 2D labels, especially when using multi-view images for inference.

[1] https://github.com/Hzzone/pytorch-openpose

NMPJPE ↓			PA-MPJPE ↓		
Single	Interact	Fusion	Single	Interact	Fusion
Self-supervised learning:					
11.17	8.28	7.75	7.22	5.42	5.40
Weakly-supervised learning (one view of the 2D ground-truth is available):					
$11.06_{\uparrow 0.11}$	$7.84_{\uparrow 0.44}$	$6.84_{\uparrow 0.91}$	$6.87_{\uparrow 0.35}$	$4.49_{\uparrow 0.93}$	$4.44_{\uparrow 0.96}$

Table C. Performance comparisons of our method under self- and weak-supervised settings.

Method	Data	Backbone	PA-JE↓	PA-VE↓	F@5↑	F@15↑
Fully-Supervised Method:						
YoutubeHand [9]	FreiHAND	Res50	8.4	8.6	0.61	0.97
I2L-MeshNet [12]	FreiHAND	Res50†	7.4	7.6	0.68	0.97
METRO [11]	FreiHAND	HRNet	6.7	6.8	0.72	0.98
Tang *et al.* [17]	FreiHAND	Res50	6.7	6.7	0.72	0.98
I2UV-HandNet [2]	FreiHAND	Res50	6.7	6.9	0.71	0.98
MobRecon [3]	FreiHAND	Res50†	6.1	6.2	0.76	0.98
Ours-SV	Frei.	Res50	7.5	7.5	0.68	0.97
Weakly-Supervised Method:						
S^2HAND [4]	Frei.	EffiNet-b0	/	/	0.42	0.89
Ours-SV	Frei.	EffiNet-b0	8.5	8.6	0.61	0.97
Ours-SV	Frei.	Res50	9.8	9.9	0.55	0.95
Self-Supervised Method:						
S^2HAND [4]	Frei.	EffiNet-b0	11.8	11.9	0.48	0.92
Ours-SV	Frei.	EffiNet-b0	11.6	11.7	0.49	0.93
Ours	HanCo	EffiNet-b0	6.3	6.8	0.71	**0.99**
Ours	HanCo	Res50	**6.2**	**6.7**	**0.72**	**0.99**

Table D. Quantitative results on the FreiHAND evaluation set. The notation † denotes using a stacked backbone structure. "Our-SV" refers to training only with our single-view network.

C.5. Results for Human Pose Estimation

Our method can also be extended to self-supervised human pose estimation. Therefore, we conduct experiments on the Human3.6M dataset [6] to compare with EpipolarPose [8] and CanonPose [18]. We train our model following the training setting of CanonPose [18]. When using camera extrinsics for multi-view self-supervised learning, the NMPJPE (mm↓) for EpipolarPose, CanonPose, and ours are 76.6, 74.3, and 71.1, respectively.

C.6. Additional Quantitative Results

FreiHand. Tab. D shows more quantitative comparisons between our approach and recent fully-supervised methods. The experimental results demonstrate that our self-supervised method achieves comparable performance to

fully supervised methods [2, 3, 9, 11, 12, 17]. We also compared our method with S^2Hand [4], a hand pose estimation method in the weakly supervised setting, which uses annotated 2D labels instead of pseudo labels to estimate 3D results. The experimental results demonstrate that our method is still effective under weak supervision.

C.7. Additional Qualitative Results

As illustrated in Fig. A, our model is capable of performing inference on multiple datasets [10, 15, 22, 23].

Fig. D shows the 2D visual comparisons between OpenPose, our single-view inference results, and the ground-truth. The results demonstrate that OpenPose can obtain plausible results for those visible joints, which is essential for self-supervised learning. However, the major problem with OpenPose is that it is not robust for invisible joints. When some joints are invisible, it can predict some particularly incorrect results and tend to predict the visible joints as the invisible ones. In contrast, our model-based method with hand prior information obtains a more robust performance towards different kinds of occlusions when the multi-view self-supervised learning provides enough accurate results for supervision.

Fig. E provides more visual comparisons between our method, EpipolarPose [8], and CanonPose [18]. All these 3D predictions are obtained with the single-view inference of the models trained by multi-view self-supervised learning. Besides, for better visualization, the predictions in the images are results after alignment with the ground-truth. From the predictions from 2 viewpoints, we can see that our method can obtain more accurate 3D joints with different gestures, backgrounds, viewpoints, occlusions, and objects in hands.

Fig. F displays the visualization of our method on the testing sequence of the Assembly101 dataset. We only train a right-hand model, and the left-hand predictions are obtained using the flipped left-hand cropped images for inference. The results demonstrate that our method can be applied to more complicated situations where the available number of hands is unknown at each time step and the occlusions are severe.

Fig. G compares our multi-view inference performance with Learnable Triangulation [7] (algebraic version). All the models are trained with self-supervised learning. The predictions are aligned with the ground-truth for better visualization. The results indicate that our method can generate more plausible results with multi-view inference when the camera parameters are available.

Fig. H illustrates our cross-dataset predictions on the testing sequence of the H2O dataset. We make use of our model trained on the HanCo dataset to estimate the hand poses with images from multiple uncalibrated cameras. The results demonstrate that our method can generalize to other multi-view settings with unknown camera parameters.

Fig. I visualizes the 2D prediction comparisons between S^2HAND [4], our method, and the ground-truth on the evaluation set of the FreiHAND dataset [23]. The results of S^2HAND are obtained by their open-source code[2] with the provided pretrained weights. As shown in the images, our model using multi-view self-supervised learning on the HanCo dataset can obtain plausible single-view predictions on the FreiHAND dataset.

Fig. J presents our failure cases on the HanCo dataset. Most of our fails are predictions from samples with challenging viewpoints and severe occlusions. Moreover, the failing predictions mainly fall into two patterns. One is incorrect hand scales and centers, and the other is wrong hand poses. Since the cross-view interaction does not explicitly use the camera extrinsics, it is difficult for it to fix those predictions with incorrect scale and center. However, from those results, we can see that it can solve the incorrect hand poses to some extent.

D. Discussions

D.1. Difference between Qiu *et al*. [13] and Ours

Our cross-view interaction network differs from Qiu *et al*. [13] in various aspects. (1) Regarding motivation, our cross-view interaction is designed to generate more reliable results for self-supervision of our single-view network while [13] aims at fusing different views' heatmaps for multi-view inference. (2) In terms of representation, our cross-view interaction utilizes compact and effective joint-level features for dual-branch interaction, while [13] fuses pixel-level features along the epipolar line, which can be computationally expensive. (3) In terms of usage, our cross-view interaction does not require camera extrinsics since we fuse information in semantic joint space while [13] relies on extrinsics for finding the epipolar line to do pixel feature fusion.

[2] https://github.com/TerenceCYJ/S2HAND

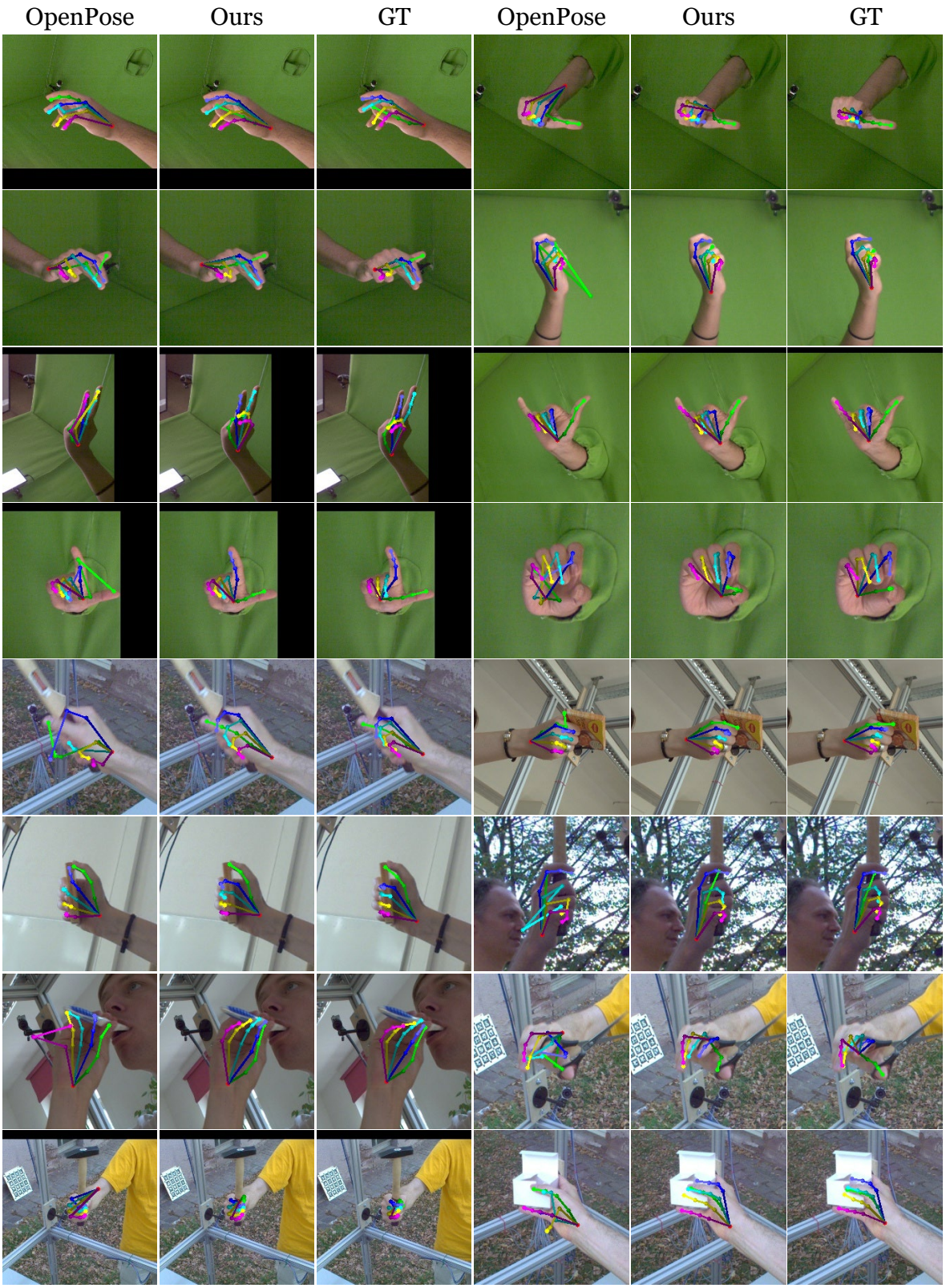

Figure D. 2D prediction (overlayed in the images) comparisons between OpenPose, ours, and the ground-truth on the HanCo dataset.

Figure E. 3D prediction comparisons between our method, EpipolarPose, and CanonPose on the HanCo dataset. Our prediction and the ground-truth are shown in solid red and dashed green respectively.

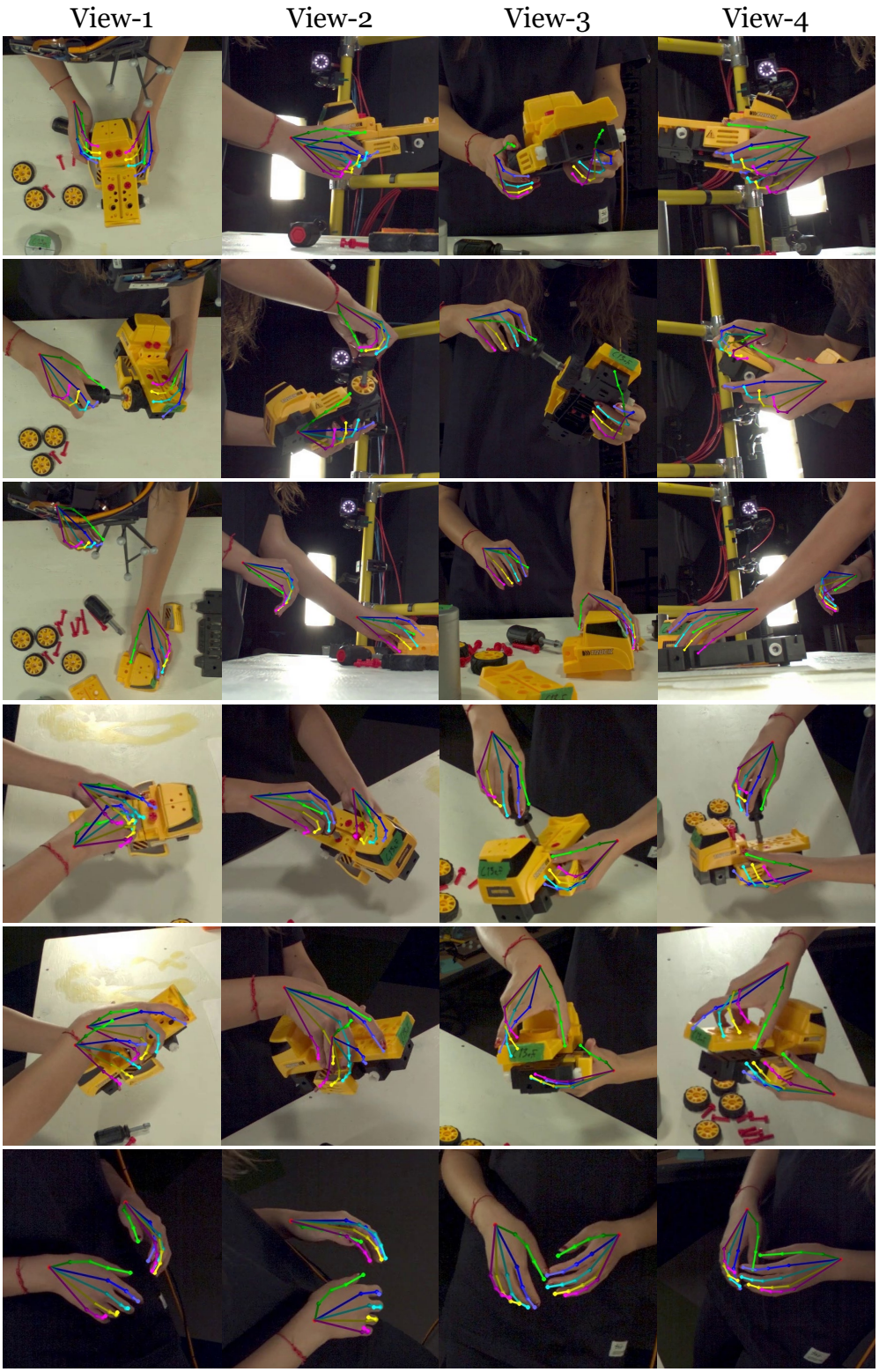

Figure F. 2D prediction (overlayed in the images) of our method in the testing sequence of the Assembly101 dataset. All the 2D image coordinates are obtained by projecting the same 3D world coordinates into different views. We utilize 8 views in total for inference. Each row shows 4 views of the projected 2D joints. The top 3 rows display the images on 4 views out of all the views, while the bottom 3 rows present the results of another 4 views.

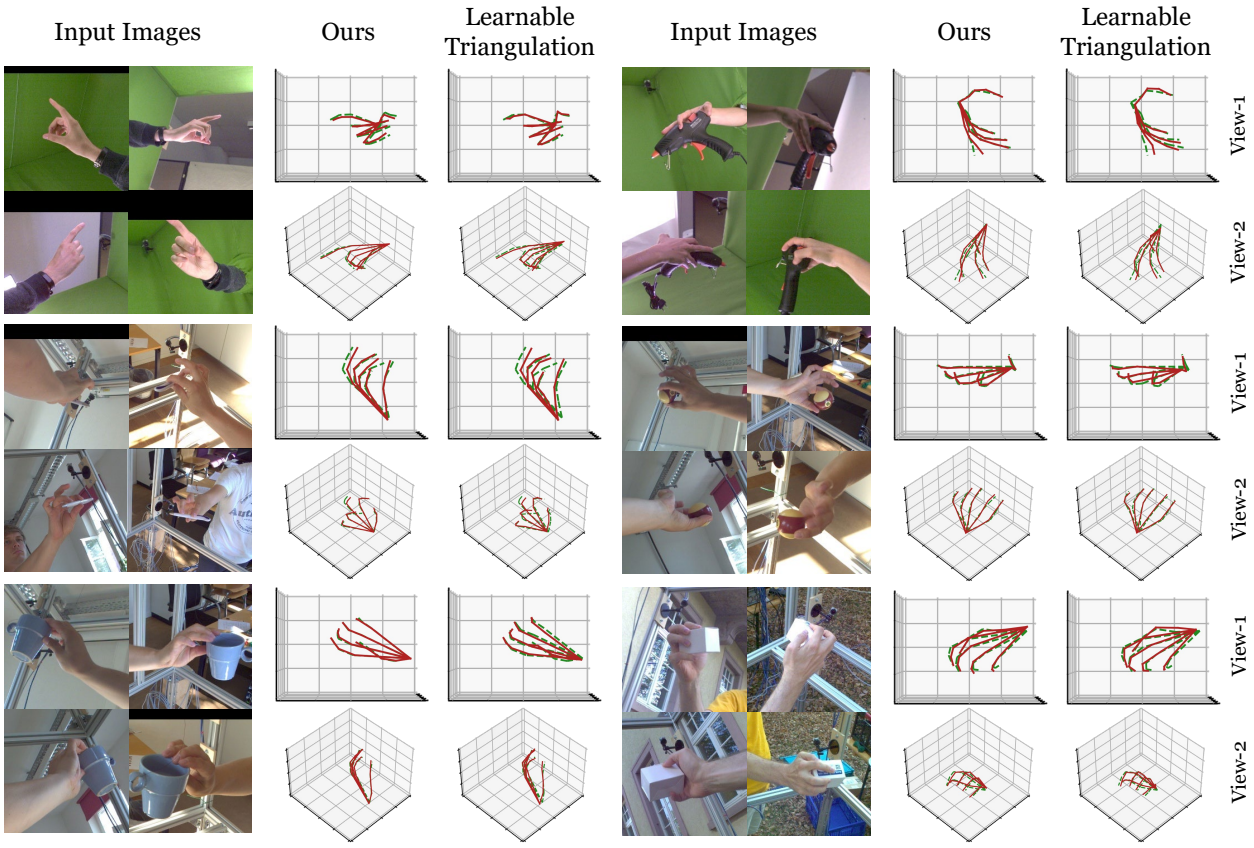

Figure G. 3D prediction comparisons between our method and Learnable Triangulation on the HanCo dataset. Our prediction and the ground-truth are shown in solid red and dashed green respectively. We use 8 views for inference and only show 4 images here.

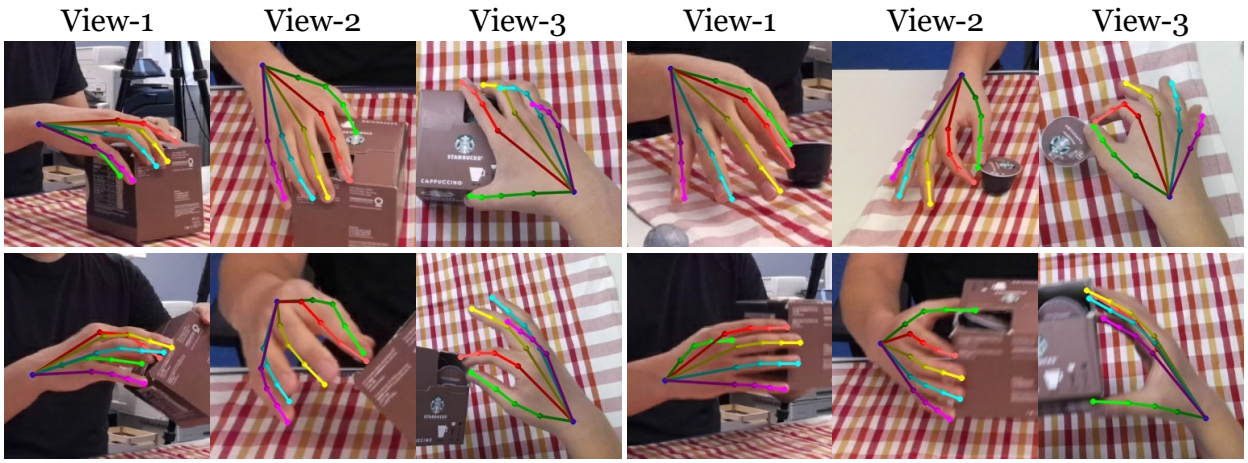

Figure H. 2D prediction (overlayed in the images) of our method in the testing sequence of the H2O dataset. The results are obtained by the model trained on the HanCo dataset. We use 3 views for inference without camera extrinsics.

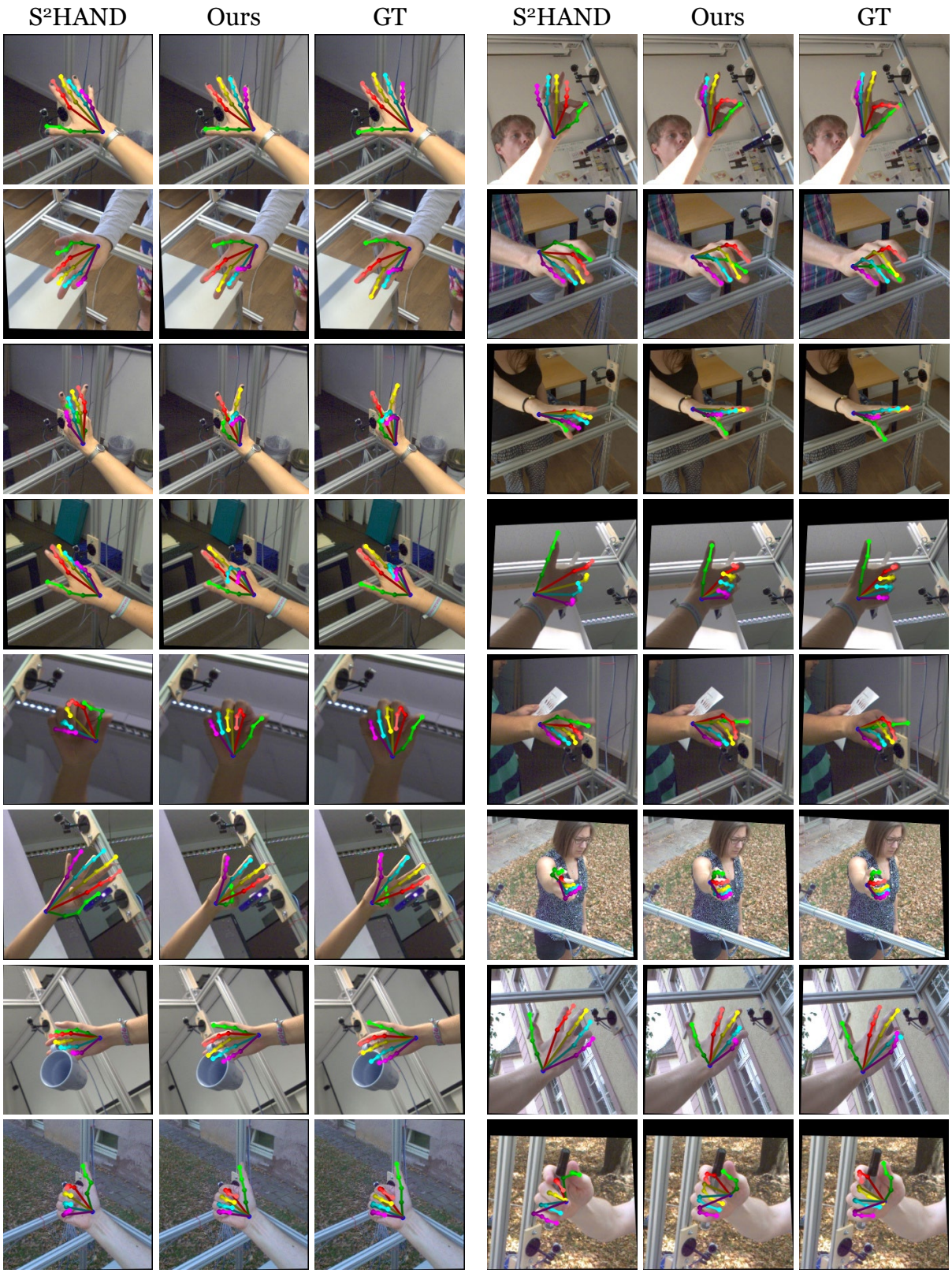

Figure I. 2D prediction (overlayed in the images) comparisons between S^2HAND, ours, and the ground-truth on the FreiHAND dataset.

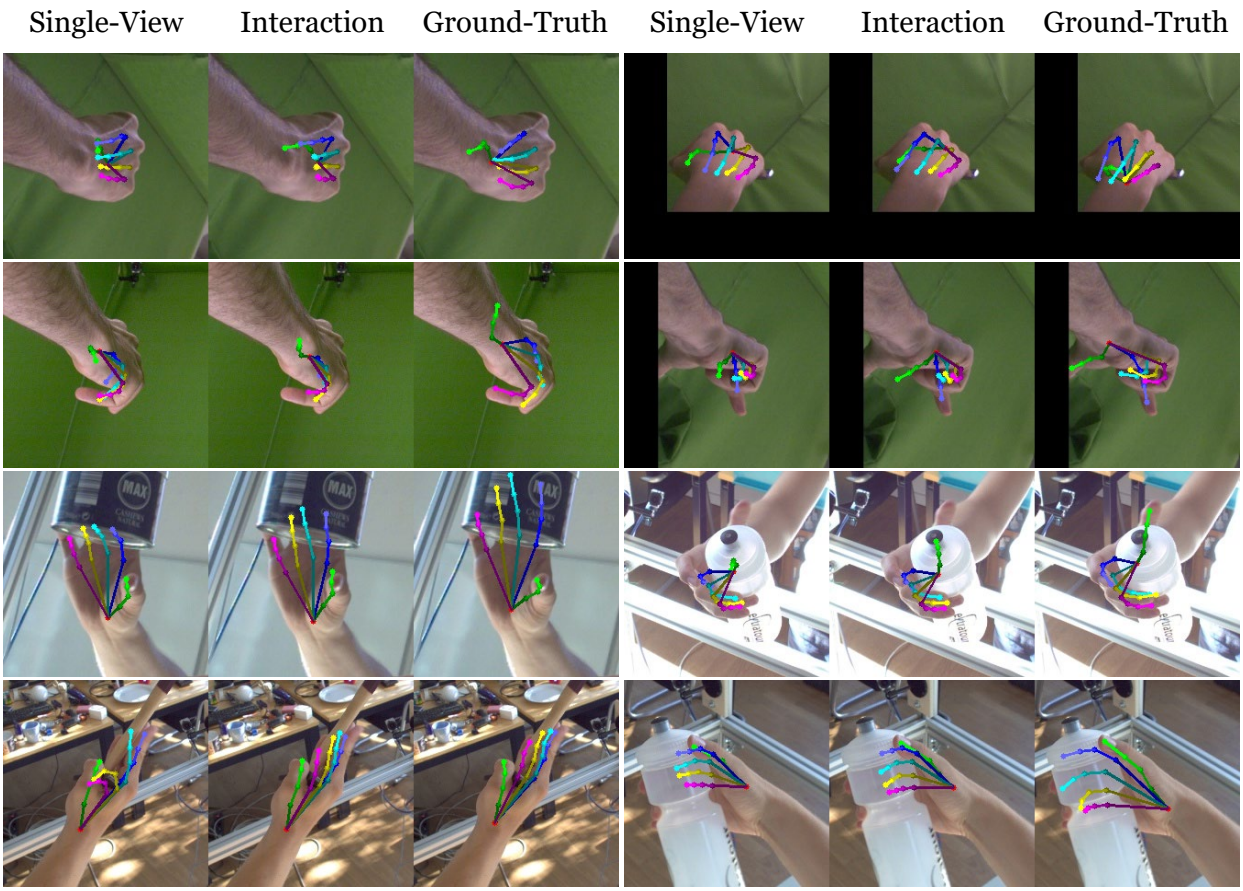

Figure J. 2D prediction (overlayed in the images) of our failure cases on the HanCo dataset. From left to right, we show our predictions from the single-view network, cross-view interaction network, and the ground-truth.

References

[1] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 3

[2] Ping Chen, Yujin Chen, Dong Yang, Fangyin Wu, Qin Li, Qingpei Xia, and Yong Tan. I2uv-handnet: Image-to-uv prediction network for accurate and high-fidelity 3d hand mesh modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12929–12938, 2021. 3, 4

[3] Xingyu Chen, Yufeng Liu, Yajiao Dong, Xiong Zhang, Chongyang Ma, Yanmin Xiong, Yuan Zhang, and Xiaoyan Guo. Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20544–20554, 2022. 3, 4

[4] Yujin Chen, Zhigang Tu, Di Kang, Linchao Bao, Ying Zhang, Xuefei Zhe, Ruizhi Chen, and Junsong Yuan. Model-based 3d hand reconstruction via self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10451–10460, 2021. 3, 4

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[6] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013. 3

[7] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7718–7727, 2019. 4

[8] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3d human pose using multi-view geometry. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1077–1086, 2019. 3, 4

[9] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4990–5000, 2020. 3, 4

[10] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10138–10148, 2021. 1, 2, 3, 4

[11] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1954–1963, 2021. 3, 4

[12] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *European Conference on Computer Vision*, pages 752–768. Springer, 2020. 3, 4

[13] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4342–4351, 2019. 4

[14] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6), 2017. 1

[15] Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21096–21106, 2022. 1, 2, 3, 4

[16] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1145–1153, 2017. 3

[17] Xiao Tang, Tianyu Wang, and Chi-Wing Fu. Towards accurate alignment in real-time 3d hand-mesh reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11698–11707, 2021. 3, 4

[18] Bastian Wandt, Marco Rudolph, Petrissa Zell, Helge Rhodin, and Bodo Rosenhahn. Canonpose: Self-supervised monocular 3d human pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13294–13304, 2021. 3, 4

[19] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018. 1

[20] Chao Wen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1042–1051, 2019. 1

[21] Xiaozheng Zheng, Pengfei Ren, Haifeng Sun, Jingyu Wang, Qi Qi, and Jianxin Liao. Sar: Spatial-aware regression for 3d hand pose and mesh reconstruction from a monocular rgb image. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 99–108. IEEE, 2021. 1

[22] Christian Zimmermann, Max Argus, and Thomas Brox. Contrastive representation learning for hand shape estimation. In *DAGM German Conference on Pattern Recognition*, pages 250–264. Springer, 2021. 1, 2, 3, 4

[23] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 813–822, 2019. 1, 4