Supplementary Material for the paper: "Reconstructing Hands in 3D with Transformers"

Georgios Pavlakos[1], Dandan Shan[2], Ilija Radosavovic[3], Angjoo Kanazawa[3], David Fouhey[4], Jitendra Malik[3]
[1]UT Austin, [2]University of Michigan, [3]UC Berkeley, [4]New York University

In this Supplementary Material we provide additional details that were not included in the main manuscript due to the limited space. We encourage the readers to also watch the Supplementary Videos:

- `video1_results.mp4` shows results of our approach on videos to highlight the temporal stability. All results are estimated on a single-frame basis, without additional smoothing.

- `video2_comparison.mp4` compares HaMeR with two state-of-the-art works for hand mesh recovery, Mesh Graphormer [14] and FrankMocap [19].

- `video3_novelview.mp4` visualizes novel views of 3D hand mesh results for HaMeR.

Besides these videos, in this document, we provide more discussion about our HInt dataset (Section S.1), and the training data we use (Section S.2). Then, we present more details about our HaMeR architecture (Section S.3) and general implementation details (Section S.4). Finally, we discuss the metrics used for evaluation (Section S.5) and present more qualitative results (Section S.6).

S.1. HInt Dataset

In this section, we discuss how we selected the frames from existing datasets to create HInt and describe the 2D hand keypoint annotation process.

S.1.1. Image sources

In HInt, we include images from three existing video resources, Hands23 [2], Epic-Kitchens [4], and Ego4D [8]. For each image, we annotate one hand. We also include annotations for sequences, where we annotate sparse frames for hand sequences of Ego4D.

For Hands23 and VISOR, I follow their original dataset splits. For Ego4D, we follow the original trainset split in our trainset but source both our valset and testset from Ego4D valset. It is because Ego4D does not release the hand-bounding box and complete hand sequences we need to use to annotate hand keypoints. Ego4D FHO splits via clip_id where we found there are similar contents spanning among different splits. To alleviate this, we split via video_id in out splits, making sure clips from the same video do not span in different splits.

Images: We select frames with hands from each of the source datasets to annotate. For Hands23, we choose from the New Days subset which are frames from YouTube videos. For Epic-Kitchens, we choose frames from the VISOR [5] benchmark. For Ego4D, we choose frames from the critical frames (pre_45, pre_30, p_15, pre-frame, contact-frame, point-of-no-return frame, and post-frame) which are defined and annotated in the FHO (Forecasting Hands and Objects) task.

In our validation and test set, we do random sampling to keep the data distribution the same as in the source datasets. In the training set, we include more challenging samples to be complementary to existing 2D keypoint datasets. Thus, for Hands23 and Epic-Kitchens, we randomly sampled half of the samples and enforced the other half to contain hand-object or hand-hand interaction. For Ego4D, we do random sampling across critical frames, since these frames typically include interactions.

Sequences: We randomly sample sequences from Ego4D FHO (Forecasting Hands and Objects) and annotate one hand for 5 critical frames (pre_45, pre_30, p_15, pre-frame ans contact-frame).

S.1.2. 2D keypoints annotation

Preparation. Annotating 2D hand keypoints from scratch is hard. In HInt, we use the help of an existing method to get rough hand keypoint estimates, and then workers are asked to refine the keypoint locations. For each hand, we feed the image and the ground-truth hand bounding box to MMPose [3] to get the initial 2D keypoint locations.

Annotation instructions. Given the Hand Keypoints Annotation Instructions (appended at the end of this document), workers refine the 21 hand keypoint locations to match the exact locations in the image. Additionally, each keypoint is annotated for "existence" and "occlusion". To the best of our knowledge, our dataset is the first one to in-

clude "occlusion" annotation for 2D hand keypoints.

S.1.3. Consistency checking

To check the annotation quality, we have a small amount of hands being annotated twice without asking the workers to check for consistency. Out of the 100 hands that were annotated twice, 90 hands were returned with valid keypoint annotations. Thus, we conducted our consistency analysis based on the 90 valid annotations in Figure S.1.

In Figure S.1 (a), we compare the offset between the two versions of annotations, normalized by the palm size (Euclidean distance from P_0 to P_9). For all visible joints, 94.6% of them are within $\times 0.25$ of palm size. In Figure S.1 (b) and (c), we checked the confusion matrix for existence and occlusion annotation and found that 100% of the existence and 90.5% of the occlusion annotation are consistent.

S.1.4. Dataset splits

Eventually, we get annotated hands from New Days (11,976), Epic-Kitchens (5,312), and Ego4D (23,174). Within the Ego4D annotated hands, we include 9,277 hands that come from 2,384 sparsely annotated sequences and could help future evaluation of temporal tasks. The detailed splits are presented in Table S.1.

Sources	Train	Val	Test
Hands23 images	9,666	550	1,760
Epic-Kitchens images	2,780	625	1,907
Ego4D images	11,652	514	1,731
Ego4D sequences	-	2,320	6,957
Total	24,098	4,009	12,355

Table S.1. Data splits for HInt.

S.2. Training data

For our training, we consolidate multiple hand datasets with 2D or 3D hand annotations. In Table S.2 we list the existing datasets we used for training, along with the number of hand examples per dataset. We also report the type of annotations of each dataset (2D or 3D), the setting they were collected (controlled multi-camera setup, synthetic or in-the-wild). For training our HaMeR model, we sample with different probabilities from each dataset, i.e., FreiHAND: 0.25, InterHand2.6M: 0.25, MTC: 0.1, HO3D: 0.05, H2O3D: 0.05, DEX YCB: 0.05, RHD: 0.05, Halpe: 0.05, COCO WholeBody: 0.1 and MPII NZSL: 0.05.

S.3. HaMeR architecture

Our HaMeR model uses a ViT-H/16 ("huge") image encoder. We start from an encoder that is pretrained on 2D body keypoint localization [22] and we finetune it for MANO [18] parameter prediction. The input image has dimension 256×192 and the ViT encoder produces 16×12 tokens, each with dimension 1280. The transformer head is a transformer decoder that takes as input a single learnable 1024-dimensional token and cross-attends to the ViT output tokens. Following the design of HMR2.0 [7], the transformer has 6 layers, hidden dimension of 1024 and 8 (64-dim) heads for self-attention and cross-attention. From the output of the transformer head, we readout the MANO and camera parameters Θ.

Method	# Examples	2D/3D	Setting
FreiHAND [26]	130k	3D	Multi-camera
InterHand2.6M [16]	1.4M	3D	Multi-camera
MTC [21]	360k	3D	Multi-camera
HO3D [9]	122k	3D	Multi-camera
H2O3D [9]	83k	3D	Multi-camera
DEX YCB [1]	400k	3D	Multi-camera
RHD [25]	62k	3D	Synthetic
Halpe [6]	34k	2D	In-the-wild
COCO WholeBody [10]	79k	2D	In-the-wild
MPII NZSL [20]	15k	2D	In-the-wild

Table S.2. Existing datasets with hand annotations that we used when training HaMeR. We list the number of hand examples per dataset, the type of annotations (2D/3D) and the setting for the data collection.

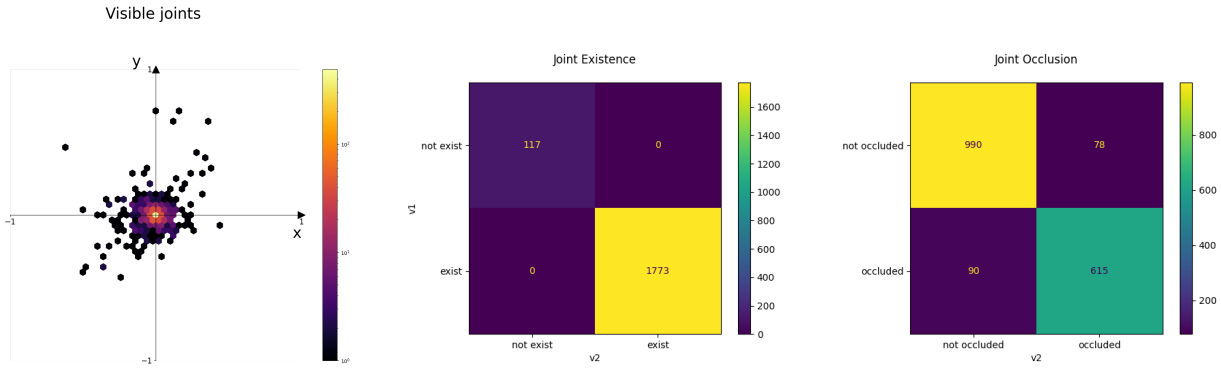
S.4. Implementation details

For training HaMeR, we use the AdamW optimizer [15] and we set the learning rate to 1e-5, the weight decay to 1e-4, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We train with an effective batch size of 1024 for 420k iterations. We perform typical augmentations during training [7], *i.e.*, scaling the bounding box, rotating, translating the center of the bounding box and applying color jitter. Our losses are balanced with different factors — 0.05 for the loss on 3D keypoints, 0.01 for the loss on 2D keypoints, 0.001 for the loss on pose parameters, 0.0005 for the loss on shape parameters and 0.0005 for the adversarial loss. For the regression target, we represent the MANO pose parameters θ using the 6D representation proposed by Zhou *et al.* [24]. Following previous work, *e.g.*, [14, 19], we train a single HaMeR model for the reconstruction of the right hand. To operate on a left hand, we apply left-right flipping to both the input image before giving it as input to the network and the output reconstruction which results to the corresponding left hand in 3D.

S.5. Metrics

In our evaluation, we use metrics that are common in the related literature.

2D Pose: We use PCK [23] to measure 2D pose accuracy.



(a) (b) (c)

Figure S.1. **Annotation consistency checking.** (a) Hexbin plot of offset between two versions of annotations. (b) Confusion matrix of existence annotation. (c) Confusion matrix of occlusion annotation.

PCK is the Percentage of Correctly localized Keypoints. We consider a keypoint to be correctly localized if its distance from the ground truth location is less than a threshold. We report PCK at different values for the threshold, @0.05, @0.1 and @0.15 of the image size.

3D Pose: To measure the accuracy of the 3D pose (3D joints), we report the PA-MPJPE [11] and AUC_J [26] metrics. PA-MPJPE measures the Mean Per Joint Position Error, *i.e.*, the average L2 error across all joints, after performing Procrustes alignment between the predicted and the ground-truth 3D Pose. AUC_J is the Area Under the Curve after computing the 3D PCK for a range of thresholds.

3D Mesh: To measure the accuracy of the 3D mesh, we report PA-MPVPE, AUC_V, F@5mm and F@15mm [26]. The first two metrics are similar to PA-MPJPE and AUC_J respectively, but the errors are computed over the reconstructed MANO vertices instead of the joints. F@5mm and F@15mm are the F-scores at two different thresholds, *i.e.*, the harmonic mean between recall and precision between two sets of points [12], the prediction and ground truth mesh. With the two thresholds, we can evaluate the accuracy at a fine and a coarse scale.

S.6. Results and analysis

In this section, we provide more quantitative and qualitative results for our approach, as well as failure cases and limitations.

S.6.1. Effect of pretraining

First, we want to evaluate the effect of the pretraining strategy for HaMeR. We use the ViT-B backbone and two pretraining strategies; pretraining on ImageNet and pretraining on the 2D human pose estimation task (after having pretrained on ImageNet first, similar to the ViTPose [22] strategy). We present the results in Table S.3, when evaluating the models on HInt. As we can see, the model pretrained on the 2D pose task consistently outperforms the model pretrained on ImageNet.

	Backbone (Pretraining)	New Days			VISOR			Ego4D		
		@0.05	@0.1	@0.15	@0.05	@0.1	@0.15	@0.05	@0.1	@0.15
All	ViT-B (ImageNet)	31.1	64.8	81.2	28.3	63.3	81.3	24.1	56.1	74.8
	ViT-B (ViTPose)	**39.6**	**72.1**	**85.4**	**37.5**	**71.6**	**86.2**	**31.8**	**63.3**	**79.2**
Vis.	ViT-B (ImageNet)	38.3	73.9	87.6	34.6	71.3	86.4	30.1	65.0	81.9
	ViT-B (ViTPose)	**49.1**	**81.8**	**91.4**	**47.0**	**81.0**	**91.7**	**41.6**	**74.9**	**87.4**
Occl.	ViT-B (ImageNet)	18.7	48.4	69.6	18.3	50.2	72.2	17.2	44.7	65.6
	ViT-B (ViTPose)	**23.5**	**54.9**	**74.5**	**23.1**	**56.9**	**77.7**	**19.3**	**48.7**	**69.4**

Table S.3. **Effect of pretraining.** We train two versions of HaMeR using different pretraining strategies, *i.e.*, pretraining on ImageNet, or on the 2D human pose estimation task (ViTPose). In both cases we use a ViT-B backbone. We report results on HInt. Pretraining on the 2D pose estimation task, outperforms vanilla ImageNet pretraining.

S.6.2. Cross-dataset generalization of HInt

Besides the vanilla version of our HaMeR model, in the main paper we show the results for a version that is trained on data from HInt. These models are evaluated on HInt. To better demonstrate the cross-dataset evaluation of HInt, we also evaluate these models on the AssemblyHands dataset [17] (since the performance on FreiHAND and HO3D is already saturated). We report our results in Table S.6.2. We observe that training on HInt improves performance on AssemblyHands, which highlights the importance of the HInt dataset.

S.6.3. Comparison with ViTPose-Hands

For further analysis, we compare our HaMeR model with the publicly available ViTPose model that is trained for the task of 2D hand keypoint detection. In Table S.5, we report results for these models on HInt. We observe that HaMeR outperforms this ViTPose baseline on the 2D metrics, while also being able to produce the full 3D shape of the hand.

	PA-MPJPE↓	MPJPE↓
Ours (no HInt)	14.3	43.5
Ours (with HInt)	**13.8**	**42.6**

Table S.4. **Effect of including HInt in training.** We evaluate two models on AssemblyHands [17]. For the first model, we do not use HInt data for training (top row), while for the second model, we use HInt data for training (second row). We observe that training on HInt improves performance on AssemblyHands.

	Method	New Days			VISOR			Ego4D		
		@0.05	@0.1	@0.15	@0.05	@0.1	@0.15	@0.05	@0.1	@0.15
All	ViTPose-Hands	32.2	51.3	61.3	40.0	64.5	75.6	23.3	41.0	52.1
	HaMeR	**49.4**	**79.3**	**89.8**	**44.4**	**77.5**	**89.7**	**40.3**	**72.4**	**85.2**
Vis.	ViTPose-Hands	44.0	62.5	70.1	55.7	77.1	83.6	35.0	52.9	61.9
	HaMeR	**62.2**	**89.0**	**95.1**	**58.5**	**88.4**	**95.0**	**53.9**	**84.2**	**91.8**
Occl.	ViTPose-Hands	13.9	32.5	46.3	21.2	46.9	63.0	10.3	26.0	38.5
	HaMeR	**28.4**	**62.4**	**80.1**	**26.9**	**61.8**	**81.2**	**24.3**	**58.7**	**77.3**

Table S.5. **Comparison of HaMeR with ViTPose-Hands.** We compare our HaMeR model with the publicly available ViTPose model for 2D hand keypoint detection. Results are presented on HInt, where HaMeR clearly outperforms the ViTPose-Hands model.

S.6.4. Qualitative results

We show additional results of HaMeR on various Internet images in Figure S.2. HaMeR returns faithful reconstructions when the hands are under heavy occlusion or in gloves, when the hands are from art paintings or from robotic hands, as well as for hands captured from both egocentric and third-person perspectives. Moreover, we provide more qualitative comparisons with state-of-the-art methods in Figure S.3. We also encourage the reader to watch the Supplementary Video, `video2_comparison.mp4` for comparisons in video form.

S.6.5. Failure cases and limitations

We show representative failure cases in Figure S.4. HaMeR may fail for input hands with extreme finger poses, unlikely appearance or hand shape, as well as extreme occlusion. Moreover, we present the effect of depending on the hand side information in Figure S.5. This dependency on hand bounding box and hand side is common in previous work [14, 19]. In our pipeline, we adopt the hand detector from Hands23 [2] to get the hand box and hand side first and then feed the hand information to HaMeR.

References

[1] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. DexYCB: A benchmark for capturing hand grasping of objects. In *CVPR*, 2021. 2

[2] Tianyi Cheng, Dandan Shan, Ayda Sultan, Jiaqi Geng, Richard EL Higgins, and David F Fouhey. Towards a richer 2D understanding of hands at scale. In *NeurIPS*, 2023. 1, 4

[3] MMPose Contributors. OpenMMLab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020. 1

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The EPIC-KITCHENS dataset. In *ECCV*, 2018. 1

[5] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. EPIC-KITCHENS VISOR benchmark: VIdeo Segmentations and Object Relations. In *NeurIPS Track on Datasets and Benchmarks*, 2022. 1

[6] Hao-Shu Fang, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. AlphaPose: Whole-body regional multi-person pose estimation and tracking in real-time. *PAMI*, 2022. 2

[7] Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4D: Reconstructing and tracking humans with transformers. In *ICCV*, 2023. 2

[8] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 1

[9] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. HOnnotate: A method for 3D annotation of hand and object poses. In *CVPR*, 2020. 2

[10] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *ECCV*, 2020. 2

[11] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 3

[12] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36 (4):1–13, 2017. 3

[13] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021. 6

[14] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, 2021. 1, 2, 4, 6, 7

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2

[16] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. InterHand2.6M: A dataset and baseline for 3D interacting hand pose estimation from a single RGB image. In *ECCV*, 2020. 2

[17] Takehiko Ohkawa, Kun He, Fadime Sener, Tomas Hodan, Luan Tran, and Cem Keskin. AssemblyHands: Towards egocentric activity understanding via 3D hand pose estimation. In *CVPR*, 2023. 3, 4

Figure S.2. **Qualitative results.** We present qualitative results of our approach on various challenging Internet images. HaMeR is particularly robust and can handle cases with heavy occlusion and interactions with objects or other hands.

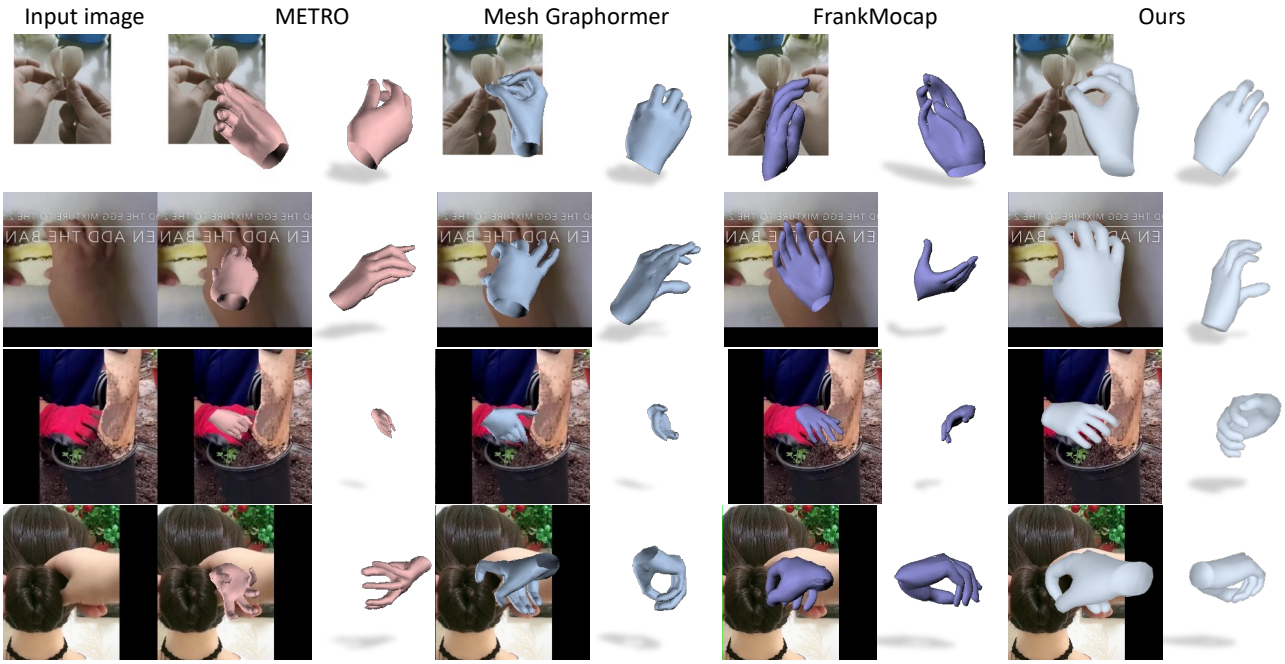


Figure S.3. **Qualitative comparison.** We compare our approach qualitatively with state-of-the-art methods for hand mesh reconstruction. The previous baselines include METRO [13], Mesh Graphormer [14] and FrankMocap [19]. We encourage the reader to also watch the Supplementary Video, `video2_comparison.mp4`, for more comparisons over time.

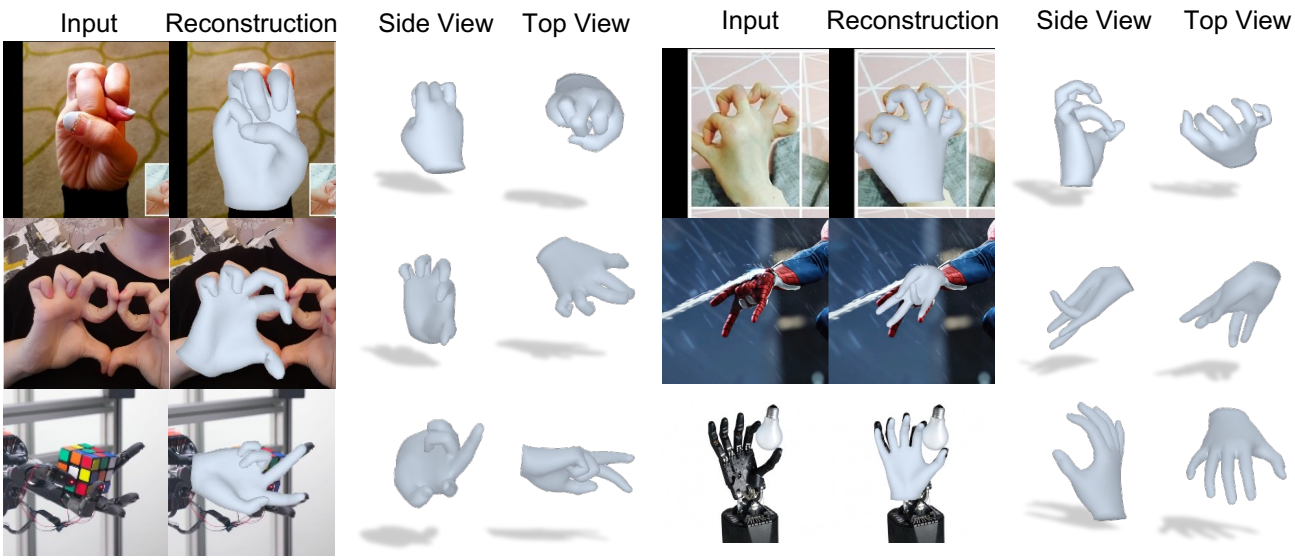


Figure S.4. **Failure cases.** We present representative failure cases of our approach. HaMeR may fail under extreme finger poses, unnatural appearance, extreme occlusions, or unnatural shape (*e.g.*, robotic hand with finger sizes that do not follow the typical human proportions).

[18] Javier Romero, Dimitris Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6), 2017. 2

[19] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. FrankMocap: A monocular 3D whole-body pose estimation system via regression and integration. In *ICCV*, 2021. 1, 2, 4, 6, 7

[20] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 2

[21] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular

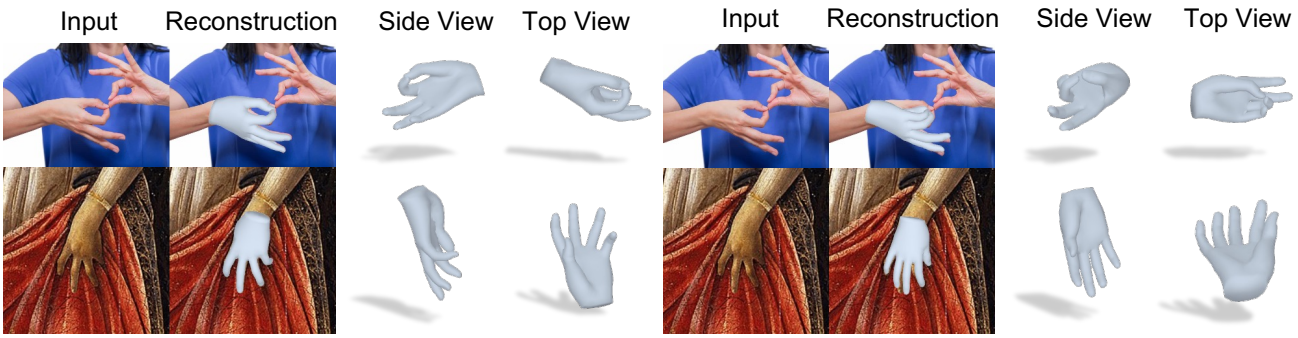


Figure S.5. **Effect of hand side information.** Similar to prior work [14, 19], HaMeR requires the hand side (left/right) information for the input image. When the given hand side is correct (left), the reconstructions align well with the 2D hands; when the given hand side is incorrect (right), the reconstructions are expected to be incorrect (*i.e.*, since the model reconstructs a hand of the opposite side), but HaMeR often returns a reasonable interpretation of the input image.

total capture: Posing face, body, and hands in the wild. In *CVPR*, 2019. 2

[22] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose: Simple vision transformer baselines for human pose estimation. *NeurIPS*, 2022. 2, 3

[23] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *PAMI*, 2012. 2

[24] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 2

[25] Christian Zimmermann and Thomas Brox. Learning to estimate 3D hand pose from single RGB images. In *ICCV*, 2017. 2

[26] 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 *ICCV*, 2019. 2, 3