CHORD: Category-level Hand-held Object Reconstruction via Shape Deformation
– Supplementary Material –

Kailin Li[1 ⋆] Lixin Yang[1,2 ⋆] Haoyu Zhen[1] Zenan Lin[3]
Xinyu Zhan[1] Licheng Zhong[1] Jian Xu[4] Kejian Wu[4] Cewu Lu[1,2 †]
[1]Shanghai Jiao Tong University [2]Shanghai Qi Zhi Institute [3]South China University of Technology [4]XREAL
[1]{kailinli, siriusyang, anye_zhen, kelvin34501, zlicheng, lucewu}@sjtu.edu.cn
[3]auzenanlin@mail.scut.edu.cn [4]{jianxu, kejian}@nreal.ai

A. Implementation Details

To implement CHORD's 2D deformation networks, $\mathcal{G}_\text{N}$, we follow the approach described in [12, 15], utilizing a pix2pixHD [14] network design. To implement CHORD's 3D deformation networks $\mathcal{G}_\text{S}$, we select a Multi-Layer Perceptrons (MLP) with five fully connected layers and a single skip connect as the object SDF decoder, which is similar to the decoders used in DeepSDF [10] and AlignSDF [3]. $\mathcal{G}_\text{S}$ takes a vector of dimensionality $\mathbb{R}^{94}$ as input. Specifically, $\mathbf{x}, \mathbf{x}^\ominus \in \mathbb{R}^3$, $\mathcal{F}_\mathcal{I} \in \mathbb{R}^{16}$, $\mathcal{F}_\text{P} \in \mathbb{R}^{48}$, $\mathcal{F}_\text{S} \in \mathbb{R}^{16}$ after dimensional mapping. $\mathcal{F}_\text{A}$ is a vector obtained from the separately sampled depth and normal maps of the hand and object, *i.e.* $(3 + 1) \times 2 = 8$ dimensions in total.

During inference, we implement a coarse-to-fine reconstruction strategy to secure precise shape reconstructions. The process starts by uniformly sampling 32^3 query points within a cubic space centered around the object of interest. Subsequently, the signed distance value is computed using the $\mathcal{G}_\text{S}$. For spaces yielding initial negative signed distance values, additional 64^3 points are sampled and their corresponding signed distance values are calculated. The final step involves reconstructing the object surface from those signed distances via the Marching Cubes algorithm [9].

Quantitative Evaluation. To quantitatively evaluate our experiments, we first simultaneously train the two preceding tasks (HPE and C-OPE) using the same CNN backbone for 100 epochs. Then, we train the CHORD's first-step network $\mathcal{G}_\text{N}$ for 100 epochs. We perturb the poses of the MANO [11] and object from the ground-truth data, and use these perturbed poses to generate two meshes: one is the hand mesh obtained by MANO's skinning function [11], and the other is the object-prior in the perturbed pose. We obtain four 2D feature maps via a differentiable renderer, which serves as the input for $\mathcal{G}_\text{N}$. We set the weight of the perceptual VGG loss [6] λ_VGG to 0.5. Finally, we train the CHORD's second-step network, $\mathcal{G}_\text{S}$, for 100 epochs. During training, we use the ground-truths with perturbation as inputs for $\mathcal{G}_\text{S}$, while during testing, we gather inputs from the outputs of the preceding tasks.

We use ResNet34 [5] as the backbone across all experiments. Our codebase is implemented in PyTorch. During training, we set the batch size to 32, the learning rate to 1×10^{-4} and decays to 1×10^{-5} after 70 epochs.

In-the-Wild Generalization. In the experiment of in-the-wild generalization, we train CHORD's two preceding tasks (HPE and C-OPE) separately. (1) To perform the hand pose estimation (HPE), we begin by predicting the hand's 3D joints using the Integral Pose Network [13]. We then map these 3D joints to MANO's rotations and shape parameters through the Inverse Kinematics Network (IKNet) [18]. (2) For category-level object-prior pose estimation (C-OPE), we simultaneously regress the object-prior's Decoupled Rotation axes (introduced in [2]) and center translation (as in [3]). These two tasks use two non-shared ResNet34 backbones. The datasets used to train the HPE model include FreiHAND [19], YouTube3D [8], OakInk [16], and DexYCB [1]. Likewise, we train the C-OPE model using COMIC dataset (our own), as well as OakInk and DexYCB.

Similarly, when performing the in-the-wild reconstruction, we utilize the mixture of COMIC, OakInk and DexYCB dataset to train CHORD's two steps deformation networks ($\mathcal{G}_\text{N}$ and $\mathcal{G}_\text{S}$).

The correspondence between the categories in our COMIC dataset and the objects in the YCB dataset (used by DexYCB) is as follows:

B. Experiment Setting Details

B.1. CHORD -vs- AlignSDF$_C$ and iHOI$_C$

Comparing our results directly with the original AlignSDF [3] and iHOI [17] is unfair because they are not trained in a category-level setting. Therefore, we retrain

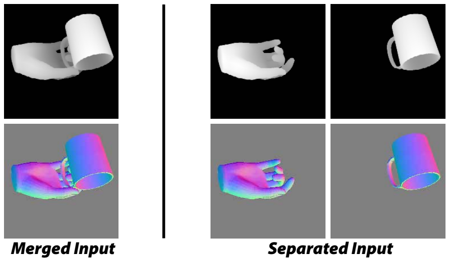

Figure 1: Ablation study on the inputs of $\mathcal{F}_{\mathrm{A}}$.

Category	YCB Object (used in DexYCB)
Bottle	002_master_chef_can, 005_tomato_soup_can, 006_mustard_bottle, 007_tuna_fish_can, 021_bleach_cleanser
Box	003_cracker_box, 004_sugar_box, 008_pudding_box, 010_potted_meat_can , 009_gelatin_box, 036_wood_block, 061_foam_brick
Mug	025_mug

Table 1: Object Category of YCB Objects in DexYCB.

both networks on our COMIC dataset. Additionally, since our model relies on the object-prior pose, we integrate the predicted pose $\mathbf{R}_{\mathbb{O}}$ and $\mathbf{t}_{\mathbb{O}}$ into AlignSDF and iHOI. Using $\mathbf{R}_{\mathbb{O}}$ and $\mathbf{t}_{\mathbb{O}}$, we transfer the query point $\mathbf{x}$ into the object prior canonical space, denoted by $\mathbf{x}^{\ominus} = \mathtt{inv}(\mathbf{R}_{\mathbb{O}}, \mathbf{t}_{\mathbb{O}}) \cdot \mathbf{x}$. We then used $\mathbf{x}^{\ominus}$ as input for both AlignSDF and iHOI networks:

$$\text{AlignSDF}_C : (\mathbf{x}, \mathbf{x}^{\ominus}, \mathcal{F}_{\mathcal{I}}(\mathbf{x})) \mapsto s(\mathbf{x}). \quad (1)$$

$$\text{iHOI}_C : (\mathbf{x}, \mathbf{x}^{\ominus}, \mathcal{F}_{\mathcal{I}}(\mathbf{x}), \mathcal{F}_{\mathrm{P}}(\mathbf{x})) \mapsto s(\mathbf{x}). \quad (2)$$

Notably, we only consider the network branch for object reconstruction.

B.2. Pose Feature of HALO network

In the main paper Sec 4.2-D: experiment settings #2, we utilized the output of the HALO [7] network as the $\mathcal{F}_{\mathrm{P}}$ in the ablation study. Specifically, HALO takes the input of the position of 21 hand joints and a query point $\mathbf{x}$, and generates an SDF value $\in \mathbb{R}^1$ which indicates the directed distance from $\mathbf{x}$ to the hand surface. We experimentally observe that incorporating the hand SDF as input assists the CHORD in avoiding surface intersections with the hand while reconstructing the object mesh. However, the accuracy of object reconstruction slightly reduces.

B.3. Inputs of the Appearance Feature

In the main paper Sec 4.2-E, we conduct an ablation study on the normal and depth maps. As shown in Fig. 1, we use Blender to render ground truth maps for minimizing the impact of noise. We train our CHORD on both the merged hand-object feature map (denoted by the green check ✓ in Table 5 of the main paper) and the separate hand and object feature maps (denoted by the blue check ✓). The results indicate a significant improvement in network performance when using the separate hand and object maps, which help mitigate the occlusion effect between the hand and object.

C. More Qualitative Evaluation

C.1. CHORD's Generalization Ability

We explore the generalization ability of CHORD under three different settings:

(1) Seen Object, Seen Domain, Unseen Camera-views (SO-SD-UC), where the objects used during testing exist in the training split from the same dataset (same domain), but observed from unseen camera viewpoint. Evaluation on OakInk and DexYCB testing sets are under this setting (Fig. 2).

(2) Seen Object, Unseen Domain, Unseen Camera-views (SO-UD-UC), where the objects used during testing exist in the training split of a different dataset (unseen domain) and are observed by unseen camera viewpoint. Evaluation on HO3D dataset is under this setting (Fig. 3).

(3) Unseen Object, Unseen Domain, Unseen Camera-views (UO-UD-UC), which is also referred to as the **zero-shot** by iHOI [17]. Evaluation on ObMan [4] and the in-the-wild testing set are under this setting (Fig. 4 and Fig. 5).

C.2. Compare with the previous SOTA

In Fig. 6, we qualitatively compare our CHORD with iHOI [17]. We use the iHOI's officially released model and code[1]. For a fair comparison, we pass the same predicted hand pose and hand-object mask to the iHOI and our CHORD model.

C.3. Examples in COMIC dataset.

We show more examples of COMIC in Fig. 7.

[1] github.com/JudyYe/ihoi

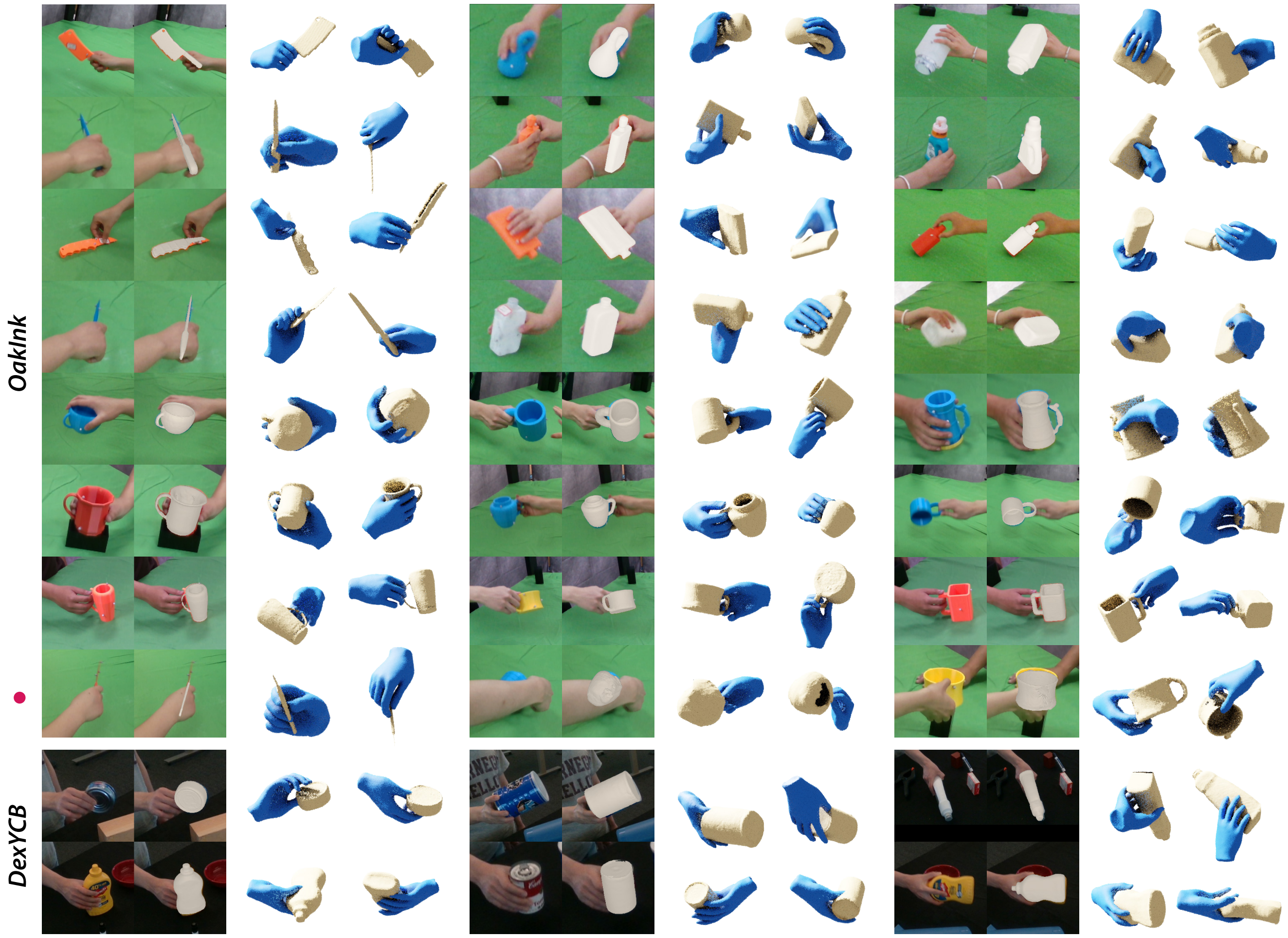

Figure 2: CHORD's results on the OakInk and DexYCB datasets under the **SO-SD-UC** setting. The line marked by a red circle ● indicates the failure cases. CHORD fails on the objects of extreme thin-wall and severe occlusions.

Figure 3: The CHORD's results on the HO3D dataset under the **SO-UD-UC** setting.

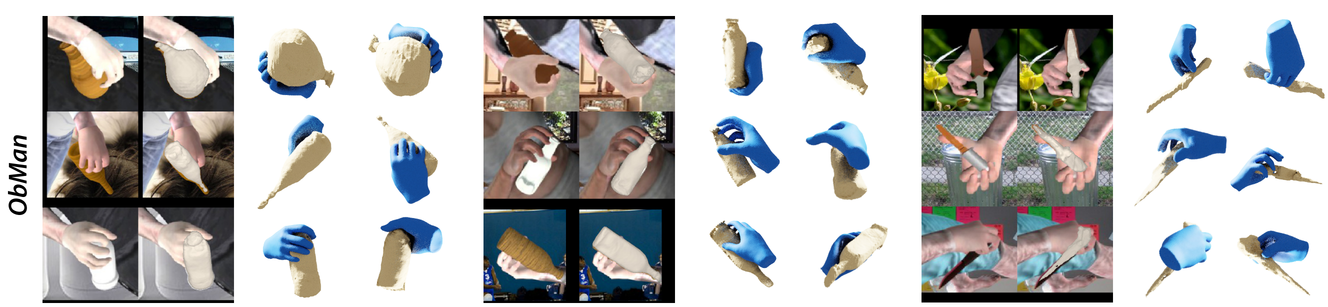

Figure 4: The CHORD's results on the synthetic dataset, ObMan, under the **UO-UD-UC** setting.

Figure 5: More examples of CHORD's performance on the in-the wild (UO-UD-UC) images. The first rows of bottle and mug categories show that our method is capable of accurately reconstructing transparent objects.

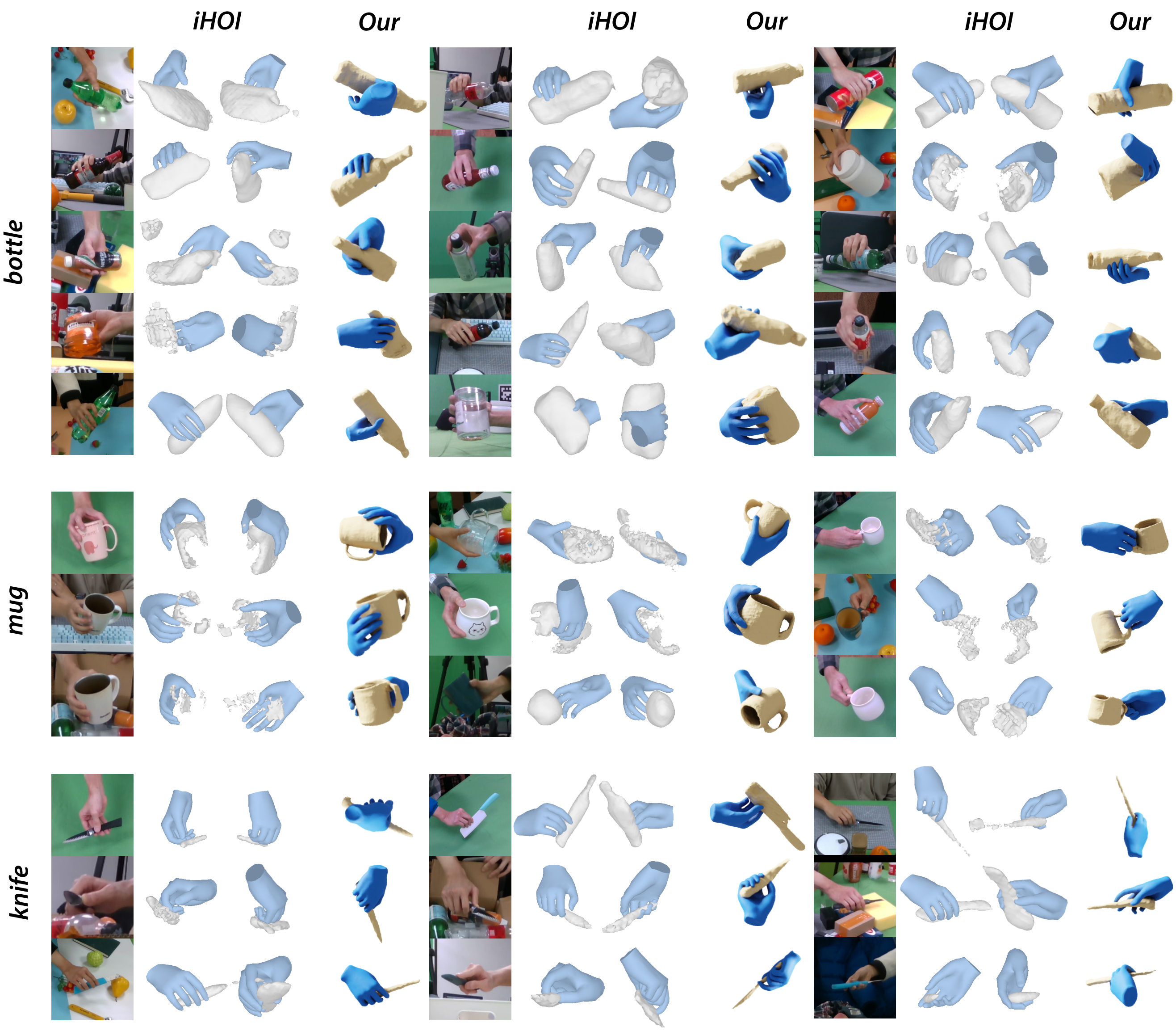

Figure 6: Comparison between our CHORD and the previous state-of-the-art, iHOI [17].

Figure 7: Examples of the six categories in our COMIC dataset.

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 *Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[2] Wei Chen, Xi Jia, Hyung Jin Chang, Jinming Duan, Linlin Shen, and Ales Leonardis. FS-Net: Fast shape-based network for category-level 6d object pose estimation with decoupled rotation mechanism. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[3] Zerui Chen, Yana Hasson, Cordelia Schmid, and Ivan Laptev. Alignsdf: Pose-aligned signed distance fields for hand-object reconstruction. In *European Conference on Computer Vision (ECCV)*, 2022. 1

[4] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[5] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[6] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[7] Korrawe Karunratanakul, Adrian Spurr, Zicong Fan, Otmar Hilliges, and Siyu Tang. A skeleton-driven neural occupancy representation for articulated hands. In *International Conference on 3D Vision (3DV)*, 2021. 2

[8] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[9] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH computer graphics*, 1987. 1

[10] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

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

[12] Shunsuke Saito, Tomas Simon, Jason M. Saragih, and Hanbyul Joo. PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[13] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *European Conference on Computer Vision (ECCV)*, 2018. 1

[14] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[15] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J. Black. ICON: Implicit clothed humans obtained from normals. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[16] Lixin Yang, Kailin Li, Xinyu Zhan, Fei Wu, Anran Xu, Liu Liu, and Cewu Lu. OakInk: A large-scale knowledge repository for understanding hand-object interaction. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[17] Yufei Ye, Abhinav Kumar Gupta, and Shubham Tulsiani. What's in your hands? 3d reconstruction of generic objects in hands. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 5

[18] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular realtime hand shape and motion capture using multi-modal data. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[19] 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 *International Conference on Computer Vision (ICCV)*, 2019. 1