HandOS: 3D Hand Reconstruction in One Stage – Supplementary Material

Xingyu Chen[1*] Zhuheng Song[3*] Xiaoke Jiang[2] Yaoqing Hu[1] Junzhi Yu[1✉] Lei Zhang[2✉]
[1] Department of Advanced Manufacturing and Robotics, College of Engineering, Peking University
[2] International Digital Economy Academy (IDEA Research)
[3] University of Chinese Academy of Sciences
idea-research.github.io/HandOSweb

Abstract

This is the supplementary document of HandOS, including implementation details (Section I), metrics (Section II), discussion on left-right classification (Section III), detector adaption (Section IV), and HO3D results (Section V), as well as more comparison (Section VI), efficiency analysis (Section VII), and visual results (Section IX). Finally, failure cases (Section VIII) and limitations are analyzed (Section XI).

I. Implementation Details

I.1. Side tuning

As shown in Fig. I, we adopt 4-scale feature maps in the visual backbone. For each scale, we utilize 3 convolution layers for feature mapping. Finally, 4-scale mapped features form $\mathbf{F}_s$.

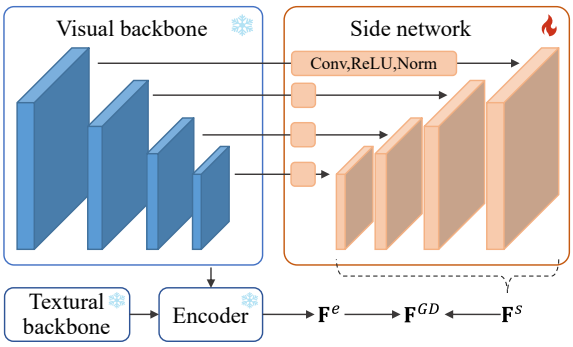

Figure I. The architecture of side tuning.

*Equal contribution. ✉Corresponding author. This work was done during Xingyu Chen's academic visit at IDEA Research and while Zhuheng Song was an intern at IDEA Research.

I.2. Loss function and Training

The full loss function is given as follows,

$$\begin{aligned}\mathcal{L} = &\lambda^{\mathbf{J}^{2D}}\mathcal{L}^{\mathbf{J}^{2D}} + \lambda^{2D}_{OKS}\mathcal{L}^{2D}_{OKS} \\ &+ \lambda^{\mathbf{V}}\mathcal{L}^{\mathbf{V}} + \lambda^{\mathbf{J}^{3D}}\mathcal{L}^{\mathbf{J}^{3D}} \\ &+ \lambda^{nomral}\mathcal{L}^{nomral} + \lambda^{edge}\mathcal{L}^{edge} \\ &+ \lambda^{\mathbf{J}^{proj}}\mathcal{L}^{\mathbf{J}^{proj}} + \lambda^{proj}_{OKS}\mathcal{L}^{proj}_{OKS} \\ &+ \lambda^{nc}\mathcal{L}^{nc},\end{aligned} \tag{I}$$

where $\lambda^{\mathbf{J}^{2D}} = \lambda^{\mathbf{J}^{3D}} = \lambda^{\mathbf{V}} = \lambda^{\mathbf{J}^{proj}} = \lambda^{edge} = 10$, $\lambda^{2D}_{OKS} = \lambda^{proj}_{OKS} = 4$, $\lambda^{normal} = 5$, $\lambda^{nc} = 0.5$.

The HandOS can be trained in an end-to-end manner with $\mathcal{L}$. To accelerate convergence and reduce experimental time, we adopt a two-stage training. First, a 2D model is trained, whose results are reported in Table 6 of the main text. The 2D model also follows the overall architecture in Fig. 2 of the main text, with all interactive layers replaced by 2D layers. Also, the 2D model does not involve query lifting and 3D vertices/camera prediction. The training data include HInt [11], COCO [8], and OneHand10K [13], with the loss function of $\lambda^{\mathbf{J}^{2D}}\mathcal{L}^{\mathbf{J}^{2D}} + \lambda^{2D}_{OKS}\mathcal{L}^{2D}_{OKS}$. The 2D training cost 3 days on 8 NVIDIA A100 GPUs.

Then, with the weights of the 2D model for initialization, we conduct our experiments on diverse benchmarks with their respective training data.

Ablation studies of loss functions are present in Table I. $\mathcal{L}_{OKS}$ improves the 2D learning efficiency from various-size instances. $\mathcal{L}^{sp} = \mathcal{L}^{normal} + \mathcal{L}^{edge}$ is crucial for structural shape learning, while $\mathcal{L}^{nc}$ is a smooth regularization. Other losses are strictly required.

II. Metrics

Percentage of correctly localized keypoints (PCK) is a metric used to evaluate the accuracy of 2D keypoint localization. A keypoint is considered correct if the distance between its predicted and ground truth locations is below a

$\mathcal{L}_{OKS}$	$\mathcal{L}^{nc}$	$\mathcal{L}^{sp}$	Ego4D$_{\text{2D-PCK}}$	FreiHand$_{\text{PV}}$
✓	✓	✓	85.3	5.6
	✓	✓	83.2	5.8
		✓	83.2	5.9
			82.9	13.2

Table I. Ablation study of loss functions.

specified threshold. We use a threshold of 0.05, 0.1, and 0.15 box size, *i.e*. PCK@0.05, PCK@0.1, and PCK@0.15.

Mean per joint/vertex position error (MPJPE/MPVPE) measures the mean per joint/vertex error by Euclidean distance (mm) between the estimated and ground-truth coordinates. Since some global variation cannot be induced from a monocular image, we use Procrustes analysis [4] to focus on local precision, *i.e*., PA-MPJPE/MPVPE.

F-score represents the harmonic mean of recall and precision calculated between two meshes with respect to a specified distance threshold. Specifically, F@5 and F@15 correspond to thresholds of 5mm and 15mm, respectively.

Area under the curve (AUC) represents the area under the PCK curve plotted against error thresholds ranging from 0 to 50mm with 100 steps.

III. Discussion on Left-Right Classification

The recognition of left and right hands is a difficult task. Previous works usually achieve this with body prior [14]. That is, the left and right are easy to understand with whole-body structure. However, there are many scenarios in which the hand appears without a body, such as in egocentric scenes. Here, the classification error increases, harming the performance of the multi-stage method.

Our one-stage pipeline is free from the impact of prior left-right information and uses the normal direction to obtain the left-right category based on the reconstructed mesh. In this manner, as long as the reconstruction results are correct, the left-right hand classification is also accurate.

Compared with the previous "left/right $\rightarrow$ mesh" paradigm, our "mesh $\rightarrow$ left/right" investigates another way for hand-side understanding. As a result, our method is superior in left-right classification. Based on the HInt test set, ViTPose [14] achieves a detection recall of 94.6% and left-right classification precision of 93.8% with its default settings. In contrast, the HandOS based on Grounding DINO reaches a detection recall of 100% (with a confidence threshold of 0.1) and left-right classification precision of 97.9%. Note that the detection precision cannot be calculated since Hint does not label all positive instances in an image.

IV. Adaptation of Other Detector

We use DINO-X [12] as the detector to build the HandOS, which achieve 0.428 box AP when measuring hand category [8] on COCO val2017 [9]. The metrics are shown in Table II, and it is evident that our HandOS is adaptable to all DETR-like detectors.

Method	New Days	VISOR	Ego4D	FreiHand
main text	75.8/75.9	85.3/85.4	85.3/85.3	5.6
w/ DINO-X	76.3/76.5	84.8/84.6	85.6/85.5	5.5

Table II. The numbers of the Hint benchmark are PCK@0.1 computed with 2D/projected joints. The numbers of FreiHand is PA-MPVPE in mm.

V. More HO3Dv3 Analysis

As explained in Fig. 5 of the main text, the inference with the ground-truth box is ill-suited, which is prevalently employed by previous work. We do not follow this setting and use the actual detection box for inference. In addition, the misaligned detection and ground truth could also induce adverse effects for HandOS training, *i.e*., query filtering based on ground truth becomes less efficient during training. Despite these unfavorable conditions, the HandOS still reaches superior results, *e.g*. 8.4 PA-MPJPE.

Also, it is necessary to evaluate the model performance with Ho3Dv3 GT boxes. As shown, although GT boxes are not involved in training, the inference can adapt to them, thanks to adaptive within-box feature localization of deformable attention, indicating our robustness to box changes.

To relieve the issue during training, we employ more training data, including FreiHand [15], HInt [11], COCO [8], OneHand10K [13], HO3Dv3 [5], DexYCB [1], CompHand [2], and H_2O3D [6]. As shown in Table III, we achieve state-of-the-art numeric results. Note that our combined training data contains 933K samples, which is smaller than that of Hamba with 2,720K samples.

Method	PJ ↓	PV ↓	F@5 ↑	F@15 ↑
AMVUR [7]	8.7	8.3	0.593	0.964
Hamba* [3]	6.9	6.8	0.681	0.982
HandOS (ours)	8.4	8.4	0.584	0.962
w/ GT box (ours)	8.4	8.5	0.581	0.962
HandOS* (ours)	**6.8**	**6.7**	**0.688**	**0.983**

Table III. Results on HO3Dv3. Errors are measured in mm. * denotes using extra training data.

VI. More Qualitative Comparison with HaMeR

More comparisons of HandOS and HaMeR are presented in Fig. II, where we are superior in accurate detection (A), novel-style adaptation (B), fine image alignment with accurate pose/shape (C, D), and reasonable occlusion awareness (E, F).

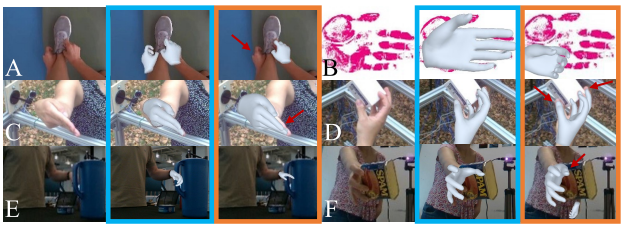

Figure II. Visual comparison between HandOS and HaMeR.

VII. Comparison of Inference Efficiency.

With P, H denoting the number of person and hand, our detector+decoder has (301+108H)G FLOPs, using 8G memory; ViTPose+HaMeR has (484P+244H)G FLOPs, using 12G memory. On RTX3090 and PyTorch, our detector takes 0.5s, and decoder time is from 0.1s (H=1) to 0.7s (H=10); VitPose+HaMeR takes (0.4+0.06P+0.1H)s.

VIII. Failure Cases

As shown in Fig. III, the HandOS could fail in false positive (the 1st row), left-right awareness (the 2nd row), inaccurate pose (the 3rd row), and geometry artifacts (the 4th row), when handling extreme lighting, occlusion, and shape conditions.

IX. Qualitative Results

Referring to Fig. IV–VII, we illustrate samples in our used datasets. As shown, the HandOS can handle various scenarios with hard poses, object occlusion, and *etc*. We also demonstrate that our HandOS is capable of real-world applications for difficult textures, shapes, lighting, and styles, as shown in Fig. VIII. The model for Fig. VIII is trained with FreiHand [15], HInt [11], CompHand [2], COCO [8], OneHand10K [13]. Note that the HandOS exhibits zero-shot generation across styles (*e.g.*, painting, cartoon), benefiting from the open-world representation of Grounding DINO [10].

X. Supplemental Video

Please refer to our homepage for dynamic results, which demonstrates frame-by-frame processing without employing any temporal strategies.

Figure III. Failure cases. Red arrows indicate errors. Samples in a triplet are input, 2D detection and joints, and 3D mesh.

XI. Limitations and Future Works

Geometry prior. The HandOS does not incorporate a geometric prior like MANO, meaning that the hand shape is learned entirely from data without relying on any predefined structural knowledge. In our opinion, incorporating an implicit prior (*e.g.*, a variational autoencoder) could accelerate the convergence of HandOS and improve the geometric realism of the predicted hand geometry.

Pose representation. We use keypoints to unify left-right hand representation. Nevertheless, obtaining a rotational pose (*i.e.* $\boldsymbol{\theta}$ in MANO) is less straightforward and requires an extra inverse kinematics module.

Temporal coherence. The HandOS is designed for single image processing without considerations for temporal coherence, which may result in jerky outputs when applied to video inference.

Future works. We plan to extend HandOS to provide versatile hand understanding. In addition to detection, 2D pose, and 3D mesh, other properties such as segmentation, texture, and object contact are also valuable considerations. Furthermore, the HandOS will be utilized to analyze human manipulation skills, contributing to advancements in embodied intelligence.

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, et al. Dexycb: A

benchmark for capturing hand grasping of objects. In *CVPR*, 2021. 2

[2] Xingyu Chen, Yufeng Liu, Dong Yajiao, Xiong Zhang, Chongyang Ma, Yanmin Xiong, Yuan Zhang, and Xiaoyan Guo. Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image. In *CVPR*, 2022. 2, 3

[3] Haoye Dong, Aviral Chharia, Wenbo Gou, Francisco Vicente Carrasco, and Fernando De la Torre. Hamba: Single-view 3d hand reconstruction with graph-guided bi-scanning mamba. In *NeurIPS*, 2024. 2

[4] John C Gower. Generalized procrustes analysis. *Psychometrika*, 1975. 2

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

[6] Shreyas Hampali, Sayan Deb Sarkar, Mahdi Rad, and Vincent Lepetit. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *CVPR*, pages 11090–11100, 2022. 2

[7] Zheheng Jiang, Hossein Rahmani, Sue Black, and Bryan M Williams. A probabilistic attention model with occlusion-aware texture regression for 3d hand reconstruction from a single rgb image. In *CVPR*, 2023. 2

[8] 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. 1, 2, 3

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 2

[10] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, 2024. 3

[11] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3D with transformers. In *CVPR*, 2024. 1, 2, 3

[12] Tianhe Ren, Yihao Chen, Qing Jiang, Zhaoyang Zeng, Yuda Xiong, Wenlong Liu, Zhengyu Ma, Junyi Shen, Yuan Gao, Xiaoke Jiang, et al. Dino-x: A unified vision model for open-world object detection and understanding. *arXiv:2411.14347*, 2024. 2

[13] Yangang Wang, Cong Peng, and Yebin Liu. Mask-pose cascaded cnn for 2d hand pose estimation from single color image. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(11):3258–3268, 2018. 1, 2, 3

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

[15] 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

Figure IV. Visualization of FreiHand evaluation set. Samples in a triplet are input, 2D detection and joints, and 3D mesh.

Figure V. Visualization of HO3Dv3 evaluation set. Samples in a triplet are input, 2D detection and joints, and 3D mesh.

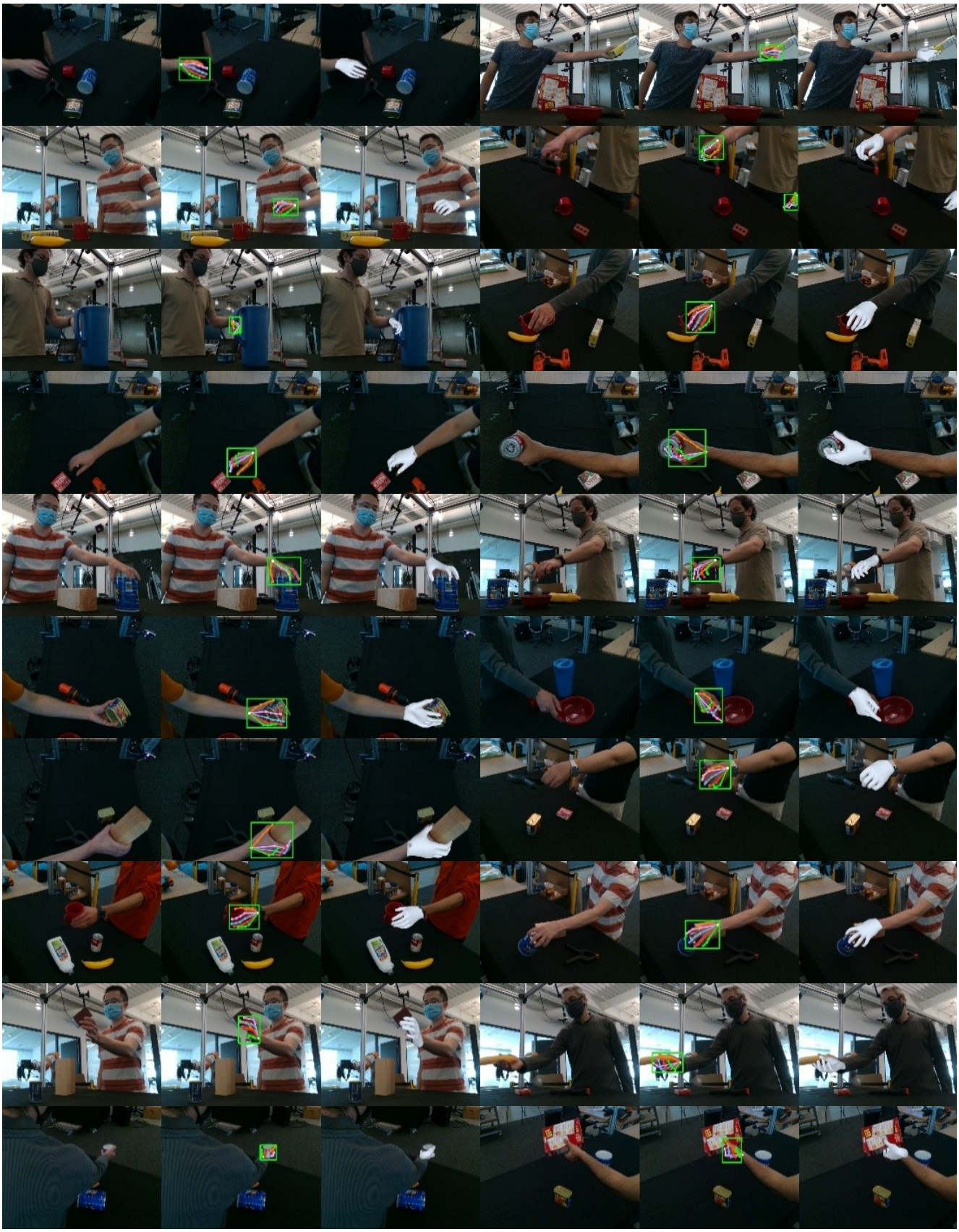

Figure VI. Visualization of DexYCB test set. Samples in a triplet are input, 2D detection and joints, and 3D mesh.

Figure VII. Visualization of HInt test set. Samples in a triplet are input, 2D detection and joints, and 3D mesh.

Figure VIII. Visualization of practical application. Samples in a triplet are input, 2D detection and joints, and 3D mesh.