arXiv:2412.01537v2 [cs.CV] 19 Mar 2025

HandOS: 3D Hand Reconstruction in One Stage

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

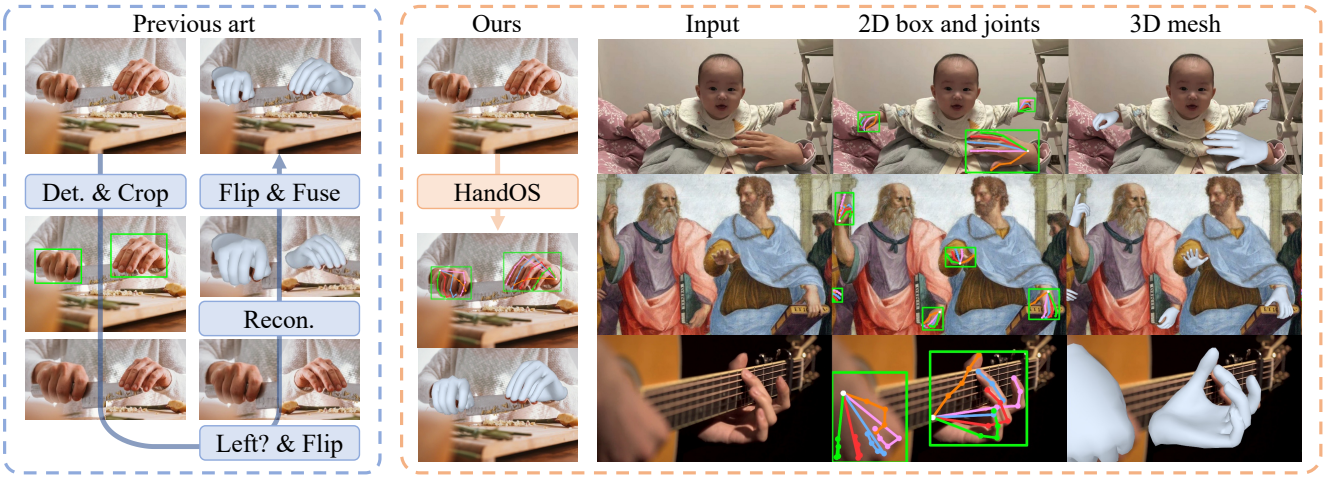

Figure 1. We present HandOS, a one-stage approach for hand reconstruction that substantially streamlines the paradigm. Additionally, we demonstrate that HandOS effectively adapts to diverse complex scenarios, making it highly applicable to real-world applications.

Abstract

Existing approaches of hand reconstruction predominantly adhere to a multi-stage framework, encompassing detection, left-right classification, and pose estimation. This paradigm induces redundant computation and cumulative errors. In this work, we propose HandOS, an end-to-end framework for 3D hand reconstruction. Our central motivation lies in leveraging a frozen detector as the foundation while incorporating auxiliary modules for 2D and 3D keypoint estimation. In this manner, we integrate the pose estimation capacity into the detection framework, while at the same time obviating the necessity of using the left-right category as a prerequisite. Specifically, we propose an interactive 2D-3D decoder, where 2D joint semantics is derived from detection cues while 3D representation is lifted from those of 2D joints. Furthermore, hierarchical attention is designed to enable the concurrent modeling of 2D joints, 3D vertices, and camera translation. Consequently, we achieve an end-to-end integration of hand detection, 2D pose estimation, and 3D mesh reconstruction within a one-stage framework, so that the above multi-stage drawbacks are overcome. Meanwhile, the HandOS reaches state-of-the-art performances on public benchmarks, e.g., 5.0 PA-MPJPE on FreiHand and 64.6% PCK@0.05 on HInt-Ego4D.

1. Introduction

The intellectual superiority of humans is expressed through their ability to use the hand to create, shape, and interact with the world. In the era of computer science and intelligence, hand understanding is crucial in reality technique [24, 57], behavior understanding [30, 46], interaction modeling [15, 69], embodied intelligence [11, 61], and *etc*.

*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.

Although hand mesh recovery has been studied for years, the pipeline is still confined to a multi-stage paradigm [2, 7, 8, 14, 36, 49], including detection, left-right recognition, and pose estimation. The necessities behind the multi-stage design are twofold. First, the hand typically occupies a limited resolution within an image, making the extraction of hand pose features from the entire image a formidable task. Hence, the detector is imperative to localize and upscale the hand regions. Second, the pose representation for the left and right hands exhibits symmetry rather than homogeneity [52]. Thus, a left-right recognizer is essential to flip the left hand to the right for a uniform pose representation. However, the multi-stage pipeline is computationally redundant, and the performance of pose estimation could be compromised by the dependencies on preceding results. For example, the error rate of detection and left-right classification reaches 11.2%, when testing ViTPose [65] on HInt test benchmark [49]. That is, some samples are determined to be incapable of yielding accurate results even before the pose estimation process. Therefore, we are inspired to overcome the above challenges by studying an end-to-end framework.

In this paper, we introduce a one-stage hand reconstruction model, termed HandOS, driven by two primary motivations. First, we utilize a pre-trained detector as the foundational model to derive the capacity of 3D reconstruction, with its parameters kept frozen during training. We choose to freeze the detector rather than simultaneously train the detection task because the approach to object detection is already well-studied, and this manner can facilitate data collection while also accelerating convergence. Moreover, to adapt the detector for our tasks, we employ a side-tuning strategy to generate adaptation features.

Second, we adopt a unified keypoint representation (*i.e.*, 2D joints and 3D vertices) for both left and right hands, instead of MANO parameters. It is known that the hand usually occupies a small portion of an entire image, so we design an instance-to-joint query expansion to extract the semantics of 2D joints from the full image guided by detection results. Then, a question naturally arises – *How to induce 3D semantics with 2D cues and perform 2D-3D information exchange?* To this end, a 2D-to-3D query lifting is proposed to transform 2D queries into 3D space. Besides, considering the different properties between 2D and 3D elements, hierarchical attention is proposed for efficient training across 2D and 3D domains. Consequently, an interactive 2D-3D decoder is formed, capable of simultaneously modeling 2D joints, 3D vertices, and camera translation.

The contribution of this work lies in three-fold. (1) First of all, we propose an end-to-end HandOS framework for 3D hand reconstruction, where pose estimation is integrated into a frozen detector. Our one-stage superiority is also demonstrated by eliminating the need for prior classification of left and right hands. Therefore, the HandOS framework offers a streamlined architecture that is well-suited for practical real-world applications. (2) We propose an interactive 2D-3D decoder with instance-to-joint query expansion, 2D-to-3D query lifting, and hierarchical attention, which allows for concurrent learning of 2D/3D keypoints and camera position. (3) The HandOS achieves superior performance in reconstruction accuracy via comprehensive evaluations and comparisons with state-of-the-art approaches, *i.e.*, 5.0, 8.4, and 5.2 PA-MPJPE on FreiHand [81], HO3Dv3 [22], and DexYCB [6] benchmarks, along with 64.6% PCK@0.05 on HInt-Ego4D [49] benchmark.

2. Related Work

3D hand reconstruction. Hand reconstruction approaches for monocular image can be broadly categorized into three types. The parametric method [1–4, 9, 25, 27, 38, 67, 69, 73, 75–78] typically employ MANO [52] as the parametric model and predicts the shape/pose coefficients to infer hand mesh. Voxel approaches [26, 44, 45, 68] utilize a 2.5D heatmap to represent 3D properties. Lastly, the vertex regression approach estimates the positions of vertices in 3D space [7, 8, 17, 33].

Recently, the transformer technique [59] has been employed to enhance the performance [14, 31, 35, 36, 49, 71, 79]. Lin *et al*. [35] leveraged the transformer to develop a vertex regression framework, where a graph network is merged with the attention mechanism for structural understanding. Pavlakos *et al*. [49] utilized the transformer in a parametric framework. Thanks to the integration of 2.7M training data, the generalization ability for in-the-wild images has been significantly enhanced. Dong *et al*. [14] also developed a transformer-like parametric framework with graph-guided Mamba [21], and a bidirectional Scan is proposed for shape-aware modeling. In our framework, we also incorporate the transformer architecture and develop an interactive 2D-3D decoder for learning keypoints in both 2D and 3D domains.

All of the aforementioned methods adhere to a multi-stage paradigm, including detection and left-right recognition. The purpose of detection is to localize the hand and resize the hand region to a fixed resolution. Rather than utilizing an external detector, we directly enhance the pre-trained detector with the capability to perform pose estimation. Besides, our pipeline eliminates the need for resizing hand regions; instead, we employ instance-to-joint query expansion and deformable attention [80] to extract hand pose features effectively. The purpose of left-right recognition is to flip hand regions, standardizing the representation of left and right hands. In contrast, our approach eliminates the need for this stage, showing that left- and right-hand data can be jointly learned using our 2D-to-3D query lifting and hierarchical attention.

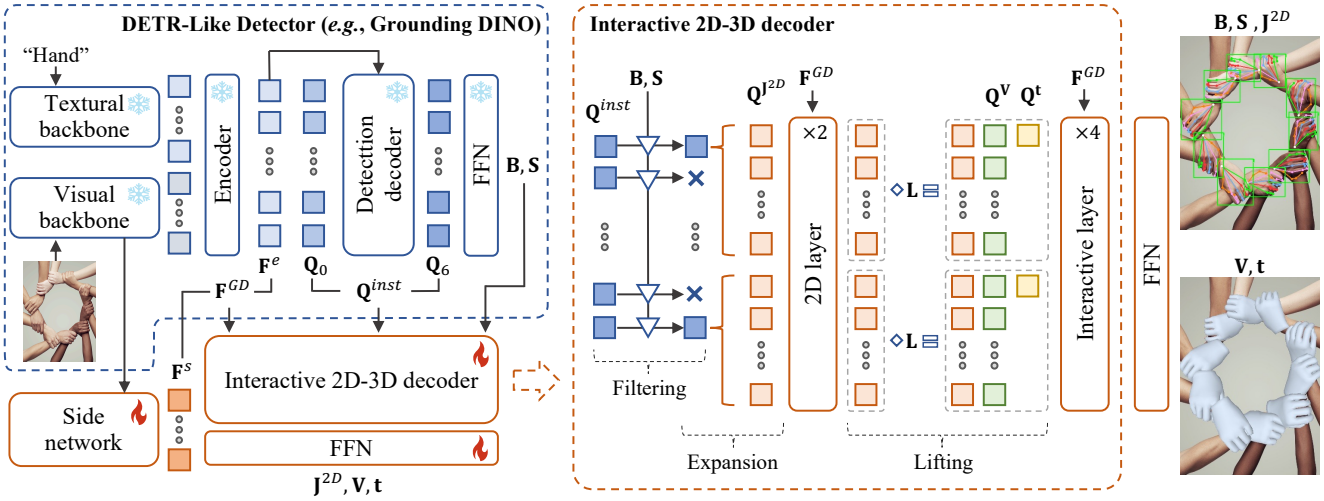

Figure 2. Overview of HandOS framework. Left: overall architecture. Right: interactive 2D-3D decoder. With off-the-shelf features, bounding boxes, and category scores from a frozen detector, the interactive 2D-3D decoder, including query filtering, expansion, lifting, and interactive layers, can understand hand pose and shape via estimating keypoints in both 2D and 3D spaces. Each query $\mathbf{Q}$ is associated with a reference box, which is not depicted in the figure for conciseness.

One-stage human pose estimation. With the advent of transformer-based object detection [5, 74], one-stage 2D pose estimation is advancing at a rapid pace. Shi *et al.* [54] proposed PETR, which is the first fully end-to-end pose estimation framework with hierarchical set prediction. Yang *et al.* [66] designed EDPose with human-to-keypoint decoder and interactive learning strategy to further enhance global and local feature aggregation.

In the field of whole-body pose estimation, several works have focused on predicting SMPLX [48] parameters from monocular images in an end-to-end fashion. For example, Sun *et al.* [56] proposed AiOS, integrating whole-body detection and pose estimation in a coarse-to-fine manner. In contrast, our approach focuses on handling images that contain only hands in a one-stage framework. In addition, instead of utilizing parametric models, we use keypoints to align the representation of left and right hands, while also unifying the representations of 2D and 3D properties.

Two-hand reconstruction. Although approaches to interaction hands can simultaneously predict the pose of two hands [34, 43, 45, 50, 60, 72], they process left and right hands using separate modules and representations. Also, they need to classify the existence of the left and right. In contrast to related works that focus on modeling hand interactions, this paper focuses on single-hand reconstruction with a unified left-right representation.

3. Method

Given a single-view image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, we aim to infer a 2D joints $\mathbf{J}^{2D} \in \mathbb{R}^{J\times 2}$, 3D vertices $\mathbf{V} \in \mathbb{R}^{V\times 3}$, and camera translation $\mathbf{t} \in \mathbb{R}^3$, where $J = 21, V = 778$. Then, 3D joints can be obtained from vertices, *i.e.*, $\mathbf{J}^{3D} = \mathcal{J}\mathbf{V}$, where $\mathcal{J}$ is the joint regressor defined by MANO [52]. With a fixed camera intrinsics $\mathbf{K}$, 3D joints can be projected into image space, *i.e.*, $\mathbf{J}^{proj} = \Pi_{\mathbf{K}}(\mathbf{J}^{3D}+\mathbf{t})$, where Π is the projection function. The overall framework is shown in Fig. 2.

3.1. Prerequisite: Grounding DINO

DETR-like detectors can serve as the foundation for HandOS. For instance, Grounding DINO [39] is utilized, which can detect objects with text prompts. In particular, we use "Hand" as the prompt without distinguishing the left and right. Referring to Fig. 2, Grounding DINO is a transformer-based architecture with a visual backbone $\mathcal{B}^v$, a textual backbone $\mathcal{B}^t$, an encoder $\mathcal{E}$, a decoder $\mathcal{D}$, and a detection head $\mathcal{H}$. The backbone [13, 16] takes images or texts as the input and produces features:

$$\mathcal{B}^v : \mathbf{I} \rightarrow \mathbf{F}^v \in \mathbb{R}^{L^v\times d^v}, \quad \mathcal{B}^t : \mathbf{T} \rightarrow \mathbf{F}^t \in \mathbb{R}^{L^t\times d^t}, \tag{1}$$

where $\mathbf{F}^v$ represents a concatenated 4-scale feature with a flattened spatial resolution. L^v, L^t denote the length of the visual/textural tokens, and d^v, d^t are token dimensions.

The encoder fuses and enhances features to generate a multi-modal representation with 6 encoding layers:

$$\mathcal{E} : (\mathbf{F}^v, \mathbf{F}^t) \rightarrow \mathbf{F}^e \in \mathbb{R}^{T^v\times d^v}. \tag{2}$$

The decoder contains 6 decoding layers, aiming at extracting features from $\mathbf{F}^e$ with deformable attention and refining queries $\mathbf{Q} \in \mathbb{R}^{Q\times d^q}$ and reference boxes $\mathbf{R} \in \mathbb{R}^{Q\times 4}$, where Q, d^q represent the number and dimension of queries. The decoding layer can be formulated as follows,

$$\mathcal{D} : (\mathbf{Q}, \mathbf{R}, \mathbf{F}^e) \rightarrow \mathbf{Q}, \quad \mathbf{R} = \texttt{FFN}(\mathbf{Q}) + \mathbf{R}, \tag{3}$$

where FFN denotes feed forward network. Finally, the detection head predicts category scores and bounding boxes:

$$\mathcal{H} : (\mathbf{Q}, \mathbf{R}) \rightarrow (\mathbf{S}, \mathbf{B}) \in \mathbb{R}^{Q \times T^t} \times \mathbb{R}^{Q \times 4}. \tag{4}$$

As a result, we collect the output in each layer, obtaining an encoding feature set $\mathcal{F}^e = \{\mathbf{F}_i^e\}_{i=1}^6$, a query set $\mathcal{Q} = \{\mathbf{Q}_i\}_{i=0}^6$. The index 0 indicates the initial elements before the decoder layers. Finally, the detection results are obtained from the last layer of detection head, producing the bounding box $\mathbf{B}$ and the corresponding score $\mathbf{S}$.

3.2. Side Tuning

To maintain off-the-shelf detection capability, we freeze all parameters in the detector. However, as the model is fully tamed for the detection task, keypoint-related representations in $\mathbf{F}^e$ remain insufficient. To conquer this difficulty, we design a learnable network with shadow layers of $\mathcal{B}^v$ as the input, generating complementary features $\mathbf{F}^s \in \mathbb{R}^{T^v \times d^v}$. As a result, the Grounding DINO provides features, *i.e.*, $\mathbf{F}^{GD} = [\mathbf{F}_6^e, \mathbf{F}^s]$, where $[\cdot, \cdot]$ denotes concatenation. Please refer to *suppl. material* for more details.

3.3. Interactive 2D-3D Decoder

The input of decoder consists of $\mathbf{F}^{GD}$, $\mathbf{B}$, $\mathbf{S}$, and queries $\mathbf{Q}^{inst} = [\mathbf{Q}_0, \mathbf{Q}_6]$, while its output includes 2D joints $\mathbf{J}^{2D}$, 3D vertices $\mathbf{V}$, and camera translation $\mathbf{t}$.

Instance query filtering. Grounding DINO produces Q instances but only a part of them belongs to the positive. During training, we employ SimOTA assigner [18] to assign instances to the positive based on ground truth. SimOTA first computes the pair-wise matching degree, which is represented by the cost $\mathbf{C}$ between the ith prediction and the jth ground truth (denoted by "$\star$"):

$$\begin{aligned}\mathbf{C}_{i,j} = &- \mathbf{S}_j^\star \log(\mathbf{S}_i) - (1 - \mathbf{S}_j^\star) \log(1 - \mathbf{S}_i) \\ &- \log(\texttt{IoU}(\mathbf{B}_i, \mathbf{B}_j^\star)).\end{aligned} \tag{5}$$

The cost function incorporates both classification error (*i.e.*, binary cross-entropy) and localization error (*i.e.*, Intersection over Union, IoU). Subsequently, an adaptive number K is derived based on IoU, and top-K instances with the lowest cost are selected as the positive samples [19]. The selected queries and boxes are denoted as $\tilde{\mathbf{Q}}^{inst} \in \mathbb{R}^{K \times d^q}$ and $\tilde{\mathbf{B}} \in \mathbb{R}^{K \times 4}$.

In the inference phase, positive queries are identified by selecting those with a score threshold T^S and a NMS threshold T^{NMS}.

Instance-to-joint query expansion. We expand instance queries for 2D joint estimation. To this end, a learnable embedding $\mathbf{e}^{\mathbf{Q}} \in \mathbb{R}^{J \times d^q}$ is designed, and joint quires are obtained by adding $\mathbf{e}^{\mathbf{Q}}$ with $\mathbf{Q}^{inst}$:

$$\mathbf{Q}^{\mathbf{J}^{2D}} \in \mathbb{R}^{K \times J \times d^q} = \tilde{\mathbf{Q}}^{inst} + \mathbf{e}^{\mathbf{Q}}. \tag{6}$$

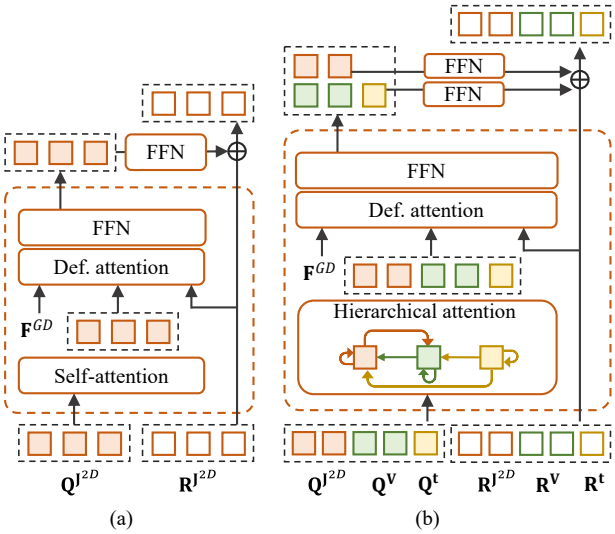

Figure 3. Decoding layers. (a) Canonical 2D layer, popularly employed by previous works. (b) Interactive layer, where hierarchical attention is designed to effectively model 2D and 3D queries.

The reference boxes for 2D joints $\mathbf{R}^{\mathbf{J}^{2D}} \in \mathbb{R}^{K \times J \times 4}$ can also be derived from instances following EDPose [66]:

$$\begin{aligned}\mathbf{R}_c^{\mathbf{J}^{2D}} &\in \mathbb{R}^{K \times J \times 2} = \texttt{FFN}(\mathbf{Q}^{\mathbf{J}^{2D}}) + \tilde{\mathbf{B}}_c, \\ \mathbf{R}_s^{\mathbf{J}^{2D}} &\in \mathbb{R}^{K \times J \times 2} = \tilde{\mathbf{B}}_s \cdot \mathbf{e}_{2D}^{\mathbf{R}},\end{aligned} \tag{7}$$

where the subscript c, s represent the center and size of the reference box, and $\mathbf{e}_{2D}^{\mathbf{R}} \in \mathbb{R}^{J \times 2}$ is the learnable embedding for the box size. FFN denotes feed forward network.

2D-to-3D query lifting. Additionally, a third query transformation is employed, namely query lifting, wherein queries and reference boxes are elevated from 2D joints to 3D vertices and camera translation. To this end, we design a learnable lifting matrix $\mathbf{L} \in \mathbb{R}^{(V+1) \times J}$ as the weights for the linear combination between 2D and 3D queries, which is initialized with MANO skinning weights. Based on $\mathbf{L}$, the lifting process can be formulated as

$$\begin{aligned}&[\mathbf{Q}^{\mathbf{V}}, \mathbf{Q}^{\mathbf{t}}] \in \mathbb{R}^{K \times (V+1) \times d^q} = \mathbf{L} \diamond \mathbf{Q}^{\mathbf{J}^{2D}}, \\ &[\mathbf{R}^{\mathbf{V}}, \mathbf{R}^{\mathbf{t}}] \in \mathbb{R}^{K \times (V+1) \times 4} = \mathbf{L} \diamond \mathbf{R}^{\mathbf{J}^{2D}}, \\ &[\mathbf{R}_c^{\mathbf{V}}, \mathbf{R}_c^{\mathbf{t}}] = \texttt{FFN}([\mathbf{Q}^{\mathbf{V}}, \mathbf{Q}^{\mathbf{t}}]) + [\mathbf{R}_c^{\mathbf{V}}, \mathbf{R}_c^{\mathbf{t}}], \\ &[\mathbf{R}_s^{\mathbf{V}}, \mathbf{R}_s^{\mathbf{t}}] = [\mathbf{R}_s^{\mathbf{V}}, \mathbf{R}_s^{\mathbf{t}}] \cdot \mathbf{e}_{3D}^{\mathbf{R}},\end{aligned} \tag{8}$$

where $\mathbf{e}_{3D}^{\mathbf{R}} \in \mathbb{R}^{(V+1) \times 2}$ is the embedding for the box size, and $\diamond$ denotes Einstein summation for linear combination.

Decoding layer and hierarchical attention. Referring to Fig. 2, the decoder comprises 6 layers, with the initial two

being designated as 2D layers, and the remaining four functioning as interactive layers. The 2D layer contains self-attention, deformable attention [80], and FFN in Fig. 3(a).

However, the popular design of Fig. 3(a) cannot directly apply to interactive 2D-3D learning. This is caused by the different properties of 2D joints, 3D vertices, and camera translation: the 2D joints and 3D vertices should exhibit invariance to translation and scale, while the camera parameters should be sensitive for both translation and scale. That is, when the object appears in different positions and scales within the image, the relative structure of the 2D joints $\mathbf{J}^{2D}$ remains unchanged, the spatial coordinates of the 3D vertices $\mathbf{V}$ stay constant, while the 3D camera translation $\mathbf{t}$ varies. Hence, $\mathbf{Q}^{\mathbf{J}^{2D}}$ and $\mathbf{Q}^{\mathbf{V}}$ should avoid performing attention operations with $\mathbf{Q}^{\mathbf{t}}$. Conversely, camera translation is significantly influenced by both 2D position and 3D geometry. Therefore, we enable $\mathbf{Q}^{\mathbf{t}}$ to focus its attention on $\mathbf{Q}^{\mathbf{J}^{2D}}$ and $\mathbf{Q}^{\mathbf{V}}$. Furthermore, 3D vertices provide a robust representation of geometric structure, whereas 2D joints capture rich semantics of image features. Therefore, we allow them to attend to each other, serving as complementary features. We refer to this operation as hierarchical attention, through which an interactive layer is formulated as shown in Fig. 3(b). The arrows in hierarchical attention indicate the visibility of the attention mechanism, with the attention mask as shown in Fig. 8(c).

Finally, we use FFN as heads for the regression of 2D joints, 3D vertices, and camera translation.

3.4. Loss Functions

2D supervision. We use point-wise L1 error and object keypoints similarity (OKS) [42] as the criterion to produce loss terms from 2D annotation (denoted by "$\star$"):

$$\begin{aligned} \mathcal{L}^{\mathbf{J}^{2D}} &= ||\mathbf{J}^{2D} - \mathbf{J}^{2D\star}||_1, \\ \mathcal{L}^{2D}_{OKS} &= \texttt{OKS}(\mathbf{J}^{2D}, \mathbf{J}^{2D\star}). \end{aligned} \tag{9}$$

3D supervision. We use point-wise L1 error, edge-wise L1 error, and normal similarity to formulate 3D loss terms:

$$\begin{aligned} &\mathcal{L}^{\mathbf{V}} = ||\mathbf{V} - \mathbf{V}^\star||_1, \quad \mathcal{L}^{\mathbf{J}^{3D}} = ||\mathbf{J}^{3D} - \mathbf{J}^{3D}||_1, \\ &\mathcal{L}^{normal} = \sum\nolimits_{\mathbf{f}\in\mathbb{F}} \sum\nolimits_{(i,j)\subset\mathbf{f}} |\frac{\mathbf{V}_i - \mathbf{V}_j}{||\mathbf{V}_i - \mathbf{V}_j||_2} \cdot \mathbf{n}^\star_{\mathbf{f}}|, \\ &\mathcal{L}^{edge} = \sum_{\mathbf{f}\in\mathbb{F}} \sum_{(i,j)\subset\mathbf{f}} |||\mathbf{V}_i - \mathbf{V}_j||_2 - ||\mathbf{V}^\star_i - \mathbf{V}^\star_j||_2|, \end{aligned} \tag{10}$$

where $\mathbb{F}$ represents mesh faces defined by MANO [52]. $\mathcal{L}^{normal}, \mathcal{L}^{edge}$ are important in our pipeline to induce a rational geometry shape without the aid of MANO inference.

Weak supervision. The majority of samples captured from daily life lack precise 3D annotations. To address this, we introduce weak loss terms based on normal consistency and projection error, enabling the use of 2D annotations for hand mesh learning:

$$\begin{aligned} \mathcal{L}^{\mathbf{J}^{proj}} &= ||\mathbf{J}^{proj} - \mathbf{J}^{2D\star}||_1, \\ \mathcal{L}^{proj}_{OKS} &= \texttt{OKS}(\mathbf{J}^{proj}, \mathbf{J}^{2D\star}), \\ \mathcal{L}^{nc} &= \sum\nolimits_{\mathbf{n}_1,\mathbf{n}_2} (1 - <\mathbf{n}_1, \mathbf{n}_2>), \end{aligned} \tag{11}$$

where $\mathbf{n}_1, \mathbf{n}_2$ are normals of neighboring faces with shared edge, and $<\cdot,\cdot>$ denotes inner product.

Overall, the total loss function is a weighted sum of the above terms, which is applied not only to the final results but also to the intermediate outputs.

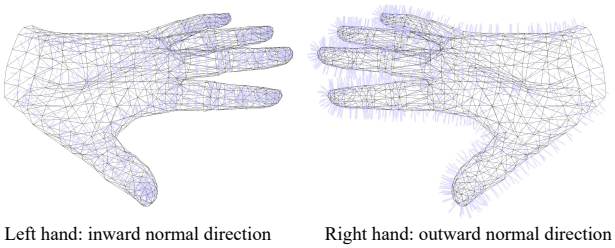

Figure 4. Normal vectors serve as left-right indicator. When applying right-hand faces to left or right vertices, the directions of the normal vectors are opposed, as illustrated by the purple lines.

3.5. Normal Vector as Left-Right Indicator

The HandOS neither requires a left-right category as a prerequisite nor explicitly incorporates a left-right classification module. Nevertheless, the left-right information is already embedded in the reconstructed mesh. Specifically, we use the normal vector as the indicator. As shown in Fig. 4, based on the right-hand face, if the mesh belongs to the left hand, the normal vectors point towards the geometric interior; otherwise, they point towards the geometric exterior. In this manner, the left-right category is obtained.

4. Experiments

4.1. Implement Details

Datasets including FreiHand [81], HO3Dv3 [22], DexYCB [6], HInt [49], COCO-WholeBody [29], and Onehand10K [62] are employed for experiments. For the FreiHand, HO3Dv3, and DexYCB benchmarks, we utilize their respective training datasets. To evaluate the HInt benchmark, we aggregate the FreiHand, HInt, COCO-WholeBody, and Onehand10K datasets for training. This combined dataset provides 204K samples, forming a subset of the 2,749K training samples used by HaMeR [49].

We utilize Grounding DINO 1.5 [51] as the pre-trained detector to exemplify our approach, noting that our framework is adaptable to other DETR-like detectors. The input

Method	PJ ↓	PV ↓	F@5 ↑	F@15 ↑
METRO [36]	6.7	6.8	0.717	0.981
MeshGraphormer [35]	5.9	6.0	0.765	0.987
MobRecon [8]	5.7	5.8	0.784	0.986
PointHMR [31]	6.1	6.6	0.720	0.984
Zhou *et al.* [79]	5.7	6.0	0.772	0.986
HaMeR [49]	6.0	5.7	0.785	0.990
Hamba [14]	5.8	5.5	0.798	**0.991**
HandOS (ours)	**5.0**	**5.3**	**0.812**	**0.991**

Table 1. Results on FreiHand. Errors are measured in mm.

Image flip in training	PJ ↓	PV ↓	F@5 ↑	F@15 ↑
✗	5.0	5.3	0.812	0.991
✓	5.3	5.6	0.799	0.989

Table 2. Results on FreiHand with left hands in training data.

is a full image, rather than a cropped hand patch, with its long edge resized to 1280 pixels, following the configuration of Grounding DINO 1.5. We employ the Adam optimizer [32] to train our model over 40 epochs with a batch size of 16. The learning rate is initialized at 0.001, with a cosine decay applied from the 25th epoch onward. On the FreiHand dataset, model training takes approximately 6 days using 8 A100-80G GPUs.

PA-MPJPE (abbreviated as PJ), PA-MPVPE (abbreviated as PV), F-socre, PCK, and AUC are used as metrics for evaluation [8, 49] with $T^S = 0.1, T^{NMS} = 0.9$.

4.2. Main Results

We use Green and Light Green to indicate the best and second results. Previous methods assume that detection and left-right category are accurate, only measuring mesh reconstruction error. In contrast, we do not use the perfect assumption, and our detector achieves 0.44 box AP when measuring hand [29] on COCO val2017 [37]. In terms of missed detection, we use $\mathbf{V} = \mathbf{0}$ for 3D metrics and set 0 for PCK/AUC. Hence, our results reflect mixed errors across detection, left-right awareness, and mesh reconstruction.

FreiHand. As shown in Table 1, the HandOS demonstrates a notable advantage over prior arts in reconstruction accuracy. Since FreiHand contains only right-hand samples, we flip images to generate left-hand samples for training. According to Table 2, we provide a unified left-right representation and support simultaneous learning for both left and right hands, delivering results comparable to those achieved with right-only training.

HO3Dv3. For the scenario of object manipulation, our method also exhibits superior performance, as shown in Table 3. However, we claim that the assumption of perfect detection made by precious works is unreasonable for HO3Dv3. Referring to Fig. 5, for highly occluded sample, only parts of the hand can be detected (*i.e.*, green box), while the ground-truth box still provides a complete hand boundary (*i.e.*, red box) that includes occluded regions. Therefore, the results reported by previous works do not accurately reflect performance in real-world applications.

Method	PJ ↓	PV ↓	F@5 ↑	F@15 ↑
AMVUR [28]	8.7	8.3	0.593	0.964
SPMHand [41]	8.8	8.6	0.574	0.962
Hamba* [14]	6.9	6.8	0.681	0.982
HandOS (ours)	8.4	8.4	0.584	0.962
HandOS* (ours)	**6.8**	**6.7**	**0.688**	**0.983**

Table 3. Results on HO3Dv3. Errors are measured in mm. * denotes using extra training data.

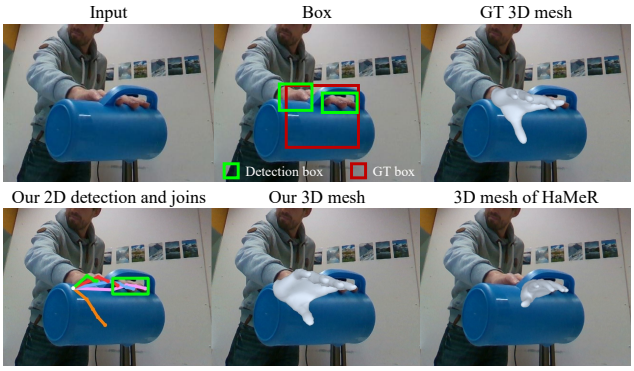

Figure 5. Visualization of HO3Dv3 with actual detection box. We claim that using GT box (red) for downstream tasks is ill-suited.

Method	PJ ↓	PV ↓	AUC ↑
Spurr *et al.* [55]	6.8	–	0.864
MobRecon [8]	6.4	5.6	–
HandOccNet [47]	5.8	5.5	–
H2ONet [64]	5.7	5.5	–
Zhou *et al.* [79]	5.5	5.5	–
HandOS (ours)	**5.2**	**5.0**	**0.896**

Table 4. Results on DexYCB. Errors are measured in mm.

In contrast, we do not rely on the assumption of perfect detection, and our one-stage pipeline can generate reasonable results from an imperfect box, as shown in Fig. 5.

DexYCB. We use DexYCB to further validate the HandOS for object manipulation. As shown in Table 4, we achieve a clear advantage in accuracy over related methods.

Hint. We utilize the HInt benchmark with New Days, VISOR, and Ego4D to evaluate HandOS on daily-life images using 2D PCK. In our method, 2D joints can be directly predicted through 2D queries or derived via 3D mesh projection. Accordingly, we report both types of PCK in Table 5. Compared to prior arts [14, 49], our training dataset is a subset of theirs, with less than one-tenth of their data size. Despite this, HandOS outperforms HaMeR and Hamba across

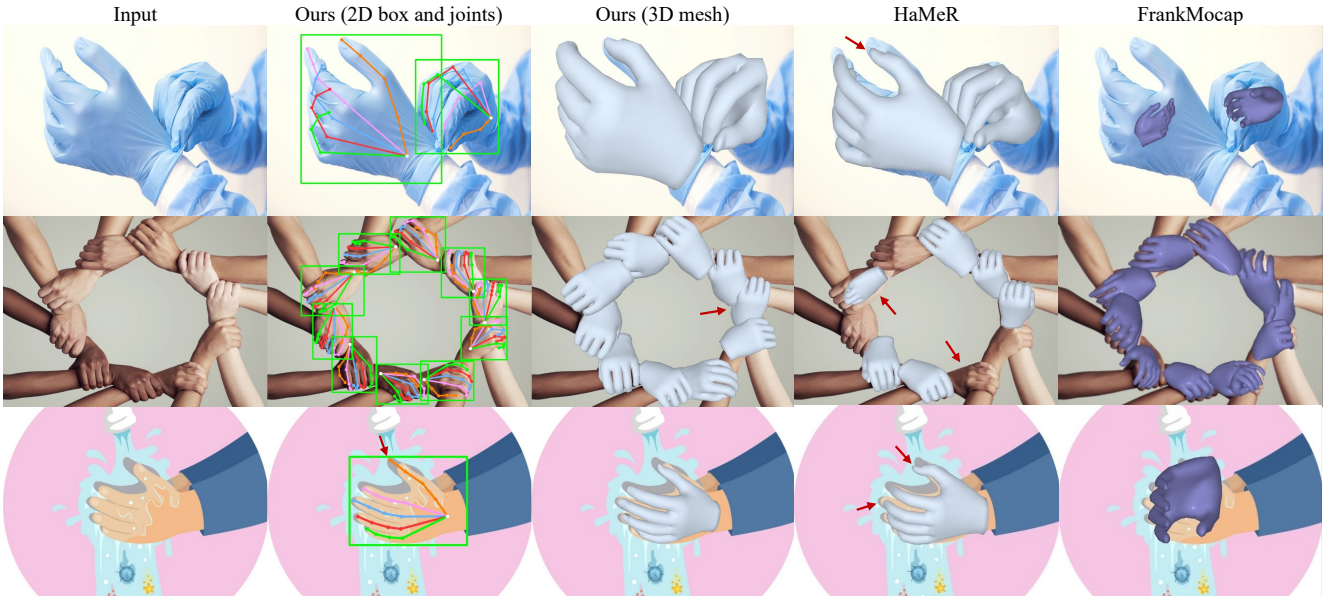

Figure 6. Visual comparison. We are adept at handling long-tail textures, crowded hands, and unseen styles. Red arrows indicate errors.

	Method	Data size	Train Hint	New Days			VISOR			Ego4D		
				@0.05	@0.1	@0.15	@0.05	@0.1	@0.15	@0.05	@0.1	@0.15
All	Hamba [14]	2,720K		48.7	79.2	90.0	47.2	80.2	91.2	–	–	–
	HaMeR [49]	2,749K	✓	51.6	**81.9**	**91.9**	56.5	**88.1**	**95.6**	46.9	79.3	90.4
	HandOS-2D (ours)	**204K**	✓	**55.8**	75.8	84.5	**66.2**	85.3	91.8	**64.6**	**85.3**	92.8
	HandOS-proj (ours)			53.7	75.9	85.1	64.8	85.4	92.0	63.4	**85.3**	**92.9**
Visible	Hamba [14]	2,720K		61.2	88.4	94.9	61.4	89.6	95.6	–	–	–
	HaMeR [49]	2,749K	✓	62.9	**89.4**	**95.8**	66.5	**92.7**	**97.4**	59.1	87.0	94.0
	HandOS-2D (ours)	**204K**	✓	**69.8**	85.0	90.6	**80.3**	92.5	95.7	**79.7**	**93.1**	**96.6**
	HandOS-proj (ours)			65.7	84.2	90.5	78.1	92.1	95.6	77.4	92.8	**96.6**
Occluded	Hamba [14]	2,720K		28.2	62.8	81.1	29.9	66.6	84.3	–	–	–
	HaMeR [49]	2,749K	✓	33.2	**68.4**	**84.8**	42.6	**79.0**	**91.3**	33.1	69.8	84.9
	HandOS-2D (ours)	**204K**	✓	35.5	63.4	76.1	**51.3**	77.9	87.4	**46.3**	75.7	86.9
	HandOS-proj (ours)			**35.8**	64.4	77.5	50.6	78.5	88.0	**46.3**	**76.1**	**87.3**

Table 5. Results on HInt. HandOS-2D and HandOS-proj denote the results from our 2D prediction and projected 3D prediction.

most PCK metrics. Notably, we achieve the highest values on all PCK@0.05 metrics, underscoring the capability for highly accurate predictions. Additionally, we obtain superior values across all metrics on HInt-Ego4D, highlighting our advantage in handling first-person perspectives.

By comparing HandOS-2D and HandOS-proj in Table 5, it can be concluded that 2D predictions perform better on visible joints, while 3D predictions excel on occluded joints, benefiting from the underlying geometric structure.

Qualitative results. As shown in Fig. 6, compared with HaMeR [49] and FrankMocap [53],the HandOS can handle complex tasks with long-tail texture, crowded objects, and unseen styles. Even without explicitly classifying left and right hands, we still achieve mostly correct results of left-right awareness and mesh reconstruction in a challenging sample, as shown in the 2nd row. Note that cartoon samples are not involved in our training. Hence, the 3rd row shows our ability to zero-shot generalization across styles.

$\mathbf{Q}^{inst}$	$\mathbf{Q}^{uni}$	$\mathbf{F}_4^e$	$\mathbf{F}_6^e$	$\mathcal{B}^v$	SwinT	New Days	VISOR	Ego4D
✓			✓	✓		**73.8**	**82.5**	**86.7**
	✓		✓	✓		71.9	80.7	86.5
✓		✓		✓		71.5	79.5	85.9
✓			✓		✓	64.7	75.6	80.1

Table 6. Ablation studies on side tuning and feature selection. The number is measured at PCK@0.1. "$\mathcal{B}^v$,SwinT" means that $\mathbf{F}^s$ is from the pre-trained visual backbone or a from-scratch SwinT.

Besides, referring to Fig. 5, HandOS can effectively handle imperfect detection results in occluded scenes. Compared with HaMeR, Fig. 5 can also reflect our one-stage superiority in eliminating cumulative errors.

4.3. Ablation Studies

On pre-trained detector. We adapt a pre-trained Grounding DINO for keypoint estimation, making it essential to in-

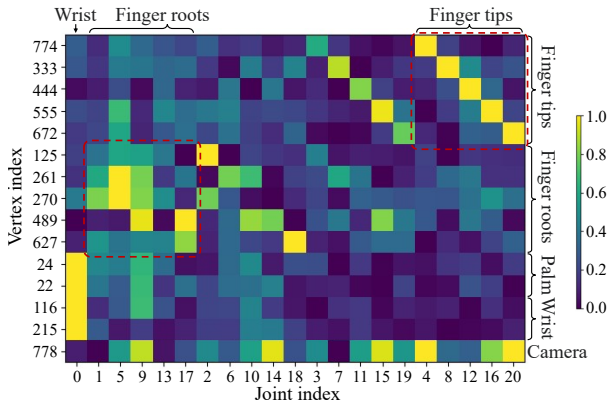

Figure 7. Query lifting matrix. Tips and roots are arranged in the order of thumb, forefinger, middle finger, ring finger, and pinky. The vertex and joint indices follow MANO and MPII orders.

vestigate how the pre-trained model aligns with the downstream task. Key configurations, including query and feature selection as well as side tuning, are examined by a 2D pose model, with their ablation studies detailed in Table 6.

We use instance queries $\mathbf{Q}^{inst} \in \mathbb{R}^{K\times d^q}$ to produce keypoint queries. In another way, keypoint queries can be shared across instances. Hence, we design a unified query $\mathbf{Q}^{uni} \in \mathbb{R}^{1\times d^q}$, which is applied to K selected instance with respective reference boxes. In addition, different from $\mathbf{Q}^{inst}$ that is given by the detector, $\mathbf{Q}^{uni}$ can be optimized along with the decoder. Referring to the first and second rows of Table 6, $\mathbf{Q}^{inst}$ has advantages over $\mathbf{Q}^{uni}$. That is, compared to $\mathbf{Q}^{uni}$, $\mathbf{Q}^{inst}$ has instance-specific information that reduces confusion among instances.

As shown in the first and third rows of Table 6, $\mathbf{F}^e_6$ outperforms $\mathbf{F}^e_4$ when used as the value for deformable attention. Since $\mathbf{F}^e_6$ is the deepest representation, it is significantly influenced by detection training, making it less optimal for keypoint estimation that demands a finer representation of object details. Nevertheless, detailed features can be supplemented through side tuning, enabling $\mathbf{F}^e_6$ that has the richest semantics to achieve superior performance.

We investigate side tuning by comparing our design with a scratch SwinT network [40]. As shown in the last rows in Table 6, an additional SwinT trained from scratch induces poor performance. This indicates that, despite being trained on the detection task, the shallow features of $\mathcal{V}$ can be mapped to adapt to other tasks, with detection pre-training also providing positive benefits.

Query lifting. A lifting matrix $\mathbf{L}$ is designed to transform 2D joint queries to 3D space. In training, $\mathbf{L}$ tends to follow a fixed pattern, and we select several typical lifting patterns for visualization. As illustrated in Fig. 7, the lifting process demonstrates semantic consistency, with the vertex queries originating from those of the corresponding joints.

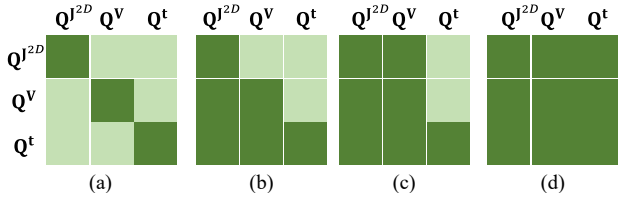

Figure 8. The ablation setting in Table 7. The dark color indicates visible attention computation.

Method	New Days	VISOR	Ego4D	FreiHand
Fig. 8(a)	75.1/70.9	84.9/81.2	84.9/82.1	6.4
Fig. 8(b)	75.7/75.8	85.1/85.3	85.2/85.2	5.7
Fig. 8(c)	**75.8/75.9**	**85.3/85.4**	**85.3/85.3**	**5.6**
Fig. 8(d)	74.6/74.8	84.1/84.2	84.6/84.5	5.9

Table 7. Ablation studies on attention strategy. The numbers of the Hint benchmark are PCK@0.1 computed with 2D/projected joints. The numbers of FreiHand is PA-MPVPE in mm.

Hierarchical attention. Considering the different properties of 2D joints, 3D vertices, and camera translation, we investigate the attention policy and demonstrate the effectiveness of our hierarchical attention, *i.e.* Fig. 8(c). Compared to it, Fig. 8(a) produces independent attention across different properties; Fig. 8(b) is unidirectional attention; and Fig. 8(d) induces a full attention policy. Referring to Table 7, our design has the best performance in terms of both 2D and 3D metrics. Fig. 8(a) results in a suboptimal 3D learning due to its invisibility to 2D quires. Fig. 8(d) make keypoints relevant to camera position, harming the relative space structure of 2D joints and 3D vertices. Fig. 8(b) has a similar performance to ours, but the interaction among keypoints is insufficient. The necessity of Fig. 8(c) is also exhibited in Table 5, where the 2D prediction is good at visible joints, while the projected estimation is adept at occluded joints. This highlights the importance of information exchange between 2D joints and 3D vertices.

5. Conclusion

We introduce HandOS, an end-to-end framework for 3D hand mesh reconstruction, which is a unified framework for hand detection, left-right awareness, and pose estimation. Additionally, we propose an interactive 2D-3D decoder with query expansion, lifting, and hierarchical attention, which supports the concurrent learning of 2D joints, 3D vertices, and camera translation. As a result, HandOS achieves state-of-the-art performance on FreiHand, Ho3Dv3, DexYCB, and HInt benchmarks.

Acknowledgment This work was partly supported by the National Natural Science Foundation of China under Grant 62403012, Grant 62233001, Grant U23B2037 and the Postdoctoral Innovative Talent Support Program under Grant BX2023004. The authors acknowledge Ling-Hao Chen for constructive discussions.

References

[1] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for RGB-based dense 3D hand pose estimation via neural rendering. In *CVPR*, 2019. 2

[2] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Weakly-supervised domain adaptation via GAN and mesh model for estimating 3D hand poses interacting objects. In *CVPR*, 2020. 2

[3] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3D hand shape and pose from images in the wild. In *CVPR*, 2019.

[4] Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. Reconstructing hand-object interactions in the wild. In *ICCV*, 2021. 2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3

[6] 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, 5, 13

[7] Xingyu Chen, Yufeng Liu, Chongyang Ma, Jianlong Chang, Huayan Wang, Tian Chen, Xiaoyan Guo, Pengfei Wan, and Wen Zheng. Camera-space hand mesh recovery via semantic aggregation and adaptive 2D-1D registration. In *CVPR*, 2021. 2

[8] 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, 6, 13, 14

[9] 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 *CVPR*, 2021. 2

[10] 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 *CVPR*, 2021.

[11] Yuanpei Chen, Yaodong Yang, Tianhao Wu, Shengjie Wang, Xidong Feng, Jiechuan Jiang, Zongqing Lu, Stephen Marcus McAleer, Hao Dong, and Song-Chun Zhu. Towards human-level bimanual dexterous manipulation with reinforcement learning. In *NeurIPS*, 2022. 1

[12] Junhyeong Cho, Kim Youwang, and Tae-Hyun Oh. Cross-attention of disentangled modalities for 3D human mesh recovery with transformers. In *ECCV*, 2022.

[13] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

[14] 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, 6, 7, 13

[15] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J Crandall. HOPE-Net: A graph-based model for hand-object pose estimation. In *CVPR*, 2020. 1

[16] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*, 2023. 3

[17] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3D hand shape and pose estimation from a single RGB image. In *CVPR*, 2019. 2

[18] Z Ge. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 4

[19] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. OTA: Optimal transport assignment for object detection. In *CVPR*, 2021. 4

[20] John C Gower. Generalized procrustes analysis. *Psychometrika*, 1975. 12

[21] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. 2023. 2

[22] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020. 2, 5, 13

[23] 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. 13

[24] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, et al. MEgATrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM TOG*, 2020. 1

[25] 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 *CVPR*, 2019. 2

[26] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5D heatmap regression. In *ECCV*, 2018. 2

[27] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. In *ICCV*, 2021. 2

[28] 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. 6, 13

[29] 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. 5, 6, 12, 13, 14

[30] Hanbyul Joo, Tomas Simon, Mina Cikara, and Yaser Sheikh. Towards social artificial intelligence: Nonverbal social signal prediction in a triadic interaction. In *CVPR*, 2019. 1

[31] Jeonghwan Kim, Mi-Gyeong Gwon, Hyunwoo Park, Hyukmin Kwon, Gi-Mun Um, and Wonjun Kim. Sampling is matter: Point-guided 3d human mesh reconstruction. In *CVPR*, 2023. 2, 6

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 6

[33] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-

supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, 2020. 2

[34] Mengcheng Li, Liang An, Hongwen Zhang, Lianpeng Wu, Feng Chen, Tao Yu, and Yebin Liu. Interacting attention graph for single image two-hand reconstruction. In *CVPR*, 2022. 3

[35] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, 2021. 2, 6

[36] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021. 2, 6

[37] 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. 6, 13

[38] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3D hand-object poses estimation with interactions in time. In *CVPR*, 2021. 2

[39] 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, 14

[40] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 8

[41] Haofan Lu, Shuiping Gou, and Ruimin Li. SPMHand: Segmentation-guided progressive multi-path 3d hand pose and shape estimation. *IEEE TMM*, 2024. 6

[42] Debapriya Maji, Soyeb Nagori, Manu Mathew, and Deepak Poddar. Yolo-Pose: Enhancing YOLO for multi person pose estimation using object keypoint similarity loss. In *CVPR*, 2022. 5

[43] Gyeongsik Moon. Bringing inputs to shared domains for 3D interacting hands recovery in the wild. In *CVPR*, 2023. 3

[44] 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 *ECCV*, 2020. 2

[45] 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, 3

[46] Evonne Ng, Shiry Ginosar, Trevor Darrell, and Hanbyul Joo. Body2Hands: Learning to infer 3D hands from conversational gesture body dynamics. In *CVPR*, 2021. 1

[47] JoonKyu Park, Yeonguk Oh, Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. HandOccNet: Occlusion-robust 3D hand mesh estimation network. In *CVPR*, pages 1496–1505, 2022. 6

[48] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3D hands, face, and body from a single image. In *CVPR*, 2019. 3

[49] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3D with transformers. In *CVPR*, 2024. 2, 5, 6, 7, 12, 13, 14

[50] Pengfei Ren, Chao Wen, Xiaozheng Zheng, Zhou Xue, Haifeng Sun, Qi Qi, Jingyu Wang, and Jianxin Liao. Decoupled iterative refinement framework for interacting hands reconstruction from a single rgb image. In *ICCV*, 2023. 3

[51] Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, Yuda Xiong, Hao Zhang, Feng Li, Peijun Tang, Kent Yu, and Lei Zhang. Grounding dino 1.5: Advance the "edge" of open-set object detection. *arXiv preprint arXiv:1810.04805*, 2024. 5

[52] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM TOG*, 2017. 2, 3, 5

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

[54] Dahu Shi, Xing Wei, Liangqi Li, Ye Ren, and Wenming Tan. End-to-end multi-person pose estimation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11069–11078, 2022. 3

[55] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3D hand pose estimation via biomechanical constraints. In *ECCV*. Springer, 2020. 6

[56] Qingping Sun, Yanjun Wang, Ailing Zeng, Wanqi Yin, Chen Wei, Wenjia Wang, Haiyi Mei, Chi-Sing Leung, Ziwei Liu, Lei Yang, et al. Aios: All-in-one-stage expressive human pose and shape estimation. In *CVPR*, 2024. 3

[57] Xiao Tang, Tianyu Wang, and Chi-Wing Fu. Towards accurate alignment in real-time 3d hand-mesh reconstruction. In *ICCV*, 2021. 1

[58] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. FastVit: A fast hybrid vision transformer using structural reparameterization. In *ICCV*, 2023.

[59] Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[60] Congyi Wang, Feida Zhu, and Shilei Wen. Memahand: Exploiting mesh-mano interaction for single image two-hand reconstruction. In *CVPR*, 2023. 3

[61] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. DexGraspNet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *ICRA*, 2023. 1

[62] 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. 5, 12, 13, 14

[63] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *ICCV*, 2019.

[64] Hao Xu, Tianyu Wang, Xiao Tang, and Chi-Wing Fu. H2ONet: Hand-occlusion-and-orientation-aware network

for real-time 3D hand mesh reconstruction. In *CVPR*, pages 17048–17058, 2023. 6

[65] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose: Simple vision transformer baselines for human pose estimation. In *NeurIPS*, 2022. 2, 12, 13

[66] Jie Yang, Ailing Zeng, Shilong Liu, Feng Li, Ruimao Zhang, and Lei Zhang. Explicit box detection unifies end-to-end multi-person pose estimation. In *ICLR*, 2023. 3, 4

[67] Lixin Yang, Jiasen Li, Wenqiang Xu, Yiqun Diao, and Cewu Lu. BiHand: Recovering hand mesh with multi-stage bisected hourglass networks. In *BMVC*, 2020. 2

[68] Linlin Yang, Shicheng Chen, and Angela Yao. SemiHand: Semi-supervised hand pose estimation with consistency. In *ICCV*, 2021. 2

[69] Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. CPF: Learning a contact potential field to model the hand-object interaction. In *ICCV*, 2021. 1, 2

[70] Lixin Yang, Kailin Li, Xinyu Zhan, Jun Lv, Wenqiang Xu, Jiefeng Li, and Cewu Lu. Artiboost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis. In *CVPR*, 2022.

[71] Yusuke Yoshiyasu. Deformable mesh transformer for 3D human mesh recovery. In *CVPR*, 2023. 2

[72] Zhengdi Yu, Shaoli Huang, Fang Chen, Toby P. Breckon, and Jue Wang. Acr: Attention collaboration-based regressor for arbitrary two-hand reconstruction. In *CVPR*, 2023. 3

[73] Baowen Zhang, Yangang Wang, Xiaoming Deng, Yinda Zhang, Ping Tan, Cuixia Ma, and Hongan Wang. Interacting two-hand 3D pose and shape reconstruction from single color image. In *ICCV*, 2021. 2

[74] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *ICLR*, 2023. 3

[75] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular RGB image. In *ICCV*, 2019. 2

[76] Xiong Zhang, Hongsheng Huang, Jianchao Tan, Hongmin Xu, Cheng Yang, Guozhu Peng, Lei Wang, and Ji Liu. Hand image understanding via deep multi-task learning. In *ICCV*, 2021.

[77] Zimeng Zhao, Xi Zhao, and Yangang Wang. TravelNet: Self-supervised physically plausible hand motion learning from monocular color images. In *ICCV*, 2021.

[78] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multi-modal data. In *CVPR*, 2020. 2

[79] Zhishan Zhou, Shihao Zhou, Zhi Lv, Minqiang Zou, Yao Tang, and Jiajun Liang. A simple baseline for efficient hand mesh reconstruction. In *CVPR*, 2024. 2, 6

[80] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2, 5

[81] 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, 5, 13, 14

Abstract

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

VI. Implementation Details

VI.1. Side tuning

As shown in Fig. IX, 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 $\mathbf{F}_s$.

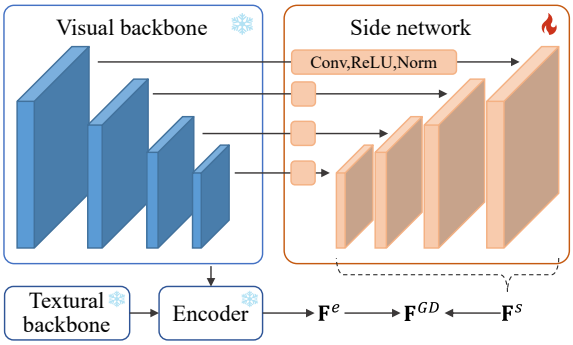

Figure IX. The architecture of side tuning.

VI.2. Loss function and Training

The full loss function is given as follows,

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

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

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 [49], COCO [29], and OneHand10K [62], 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 VIII. $\mathcal{L}_{OKS}$ 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.

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

Table VIII. Ablation study of loss functions.

VII. 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 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 [20] 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.

VIII. Discussion on Left-Right Classification

The recognition of left and right hands is a difficult task. Previous works usually achieve this with body prior [65]. 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 → mesh" paradigm, our "mesh → 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 [65] 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.

IX. Adaptation of Other Detector

We use DINO-X [**?**] as the detector to build the HandOS, which achieve 0.428 box AP when measuring hand category [29] on COCO val2017 [37]. The metrics are shown in Table IX, 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 IX. The numbers of the Hint benchmark are PCK@0.1 computed with 2D/projected joints. The numbers of FreiHand is PA-MPVPE in mm.

X. 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 [81], HInt [49], COCO [29], OneHand10K [62], HO3Dv3 [22], DexYCB [6], CompHand [8], and H_2O3D [23]. As shown in Table X, 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 [28]	8.7	8.3	0.593	0.964
Hamba* [14]	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 X. Results on HO3Dv3. Errors are measured in mm. * denotes using extra training data.

XI. More Qualitative Comparison with HaMeR

More comparisons of HandOS and HaMeR are presented in Fig. X, 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).

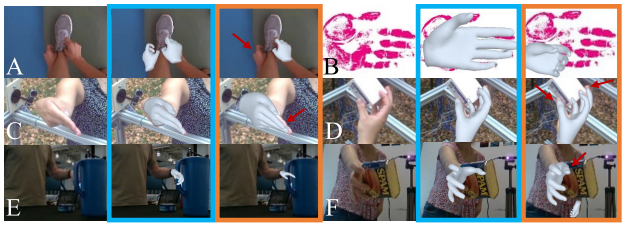

Figure X. Visual comparison between HandOS and HaMeR.

XII. 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.

XIII. Failure Cases

As shown in Fig. XI, 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.

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

XIV. Qualitative Results

Referring to Fig. XII–XV, 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. XVI. The model for Fig. XVI is trained with FreiHand [81], HInt [49], CompHand [8], COCO [29], OneHand10K [62]. Note that the HandOS exhibits zero-shot generation across styles (*e.g.*, painting, cartoon), benefiting from the open-world representation of Grounding DINO [39].

XV. Supplemental Video

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

XVI. 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.

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

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

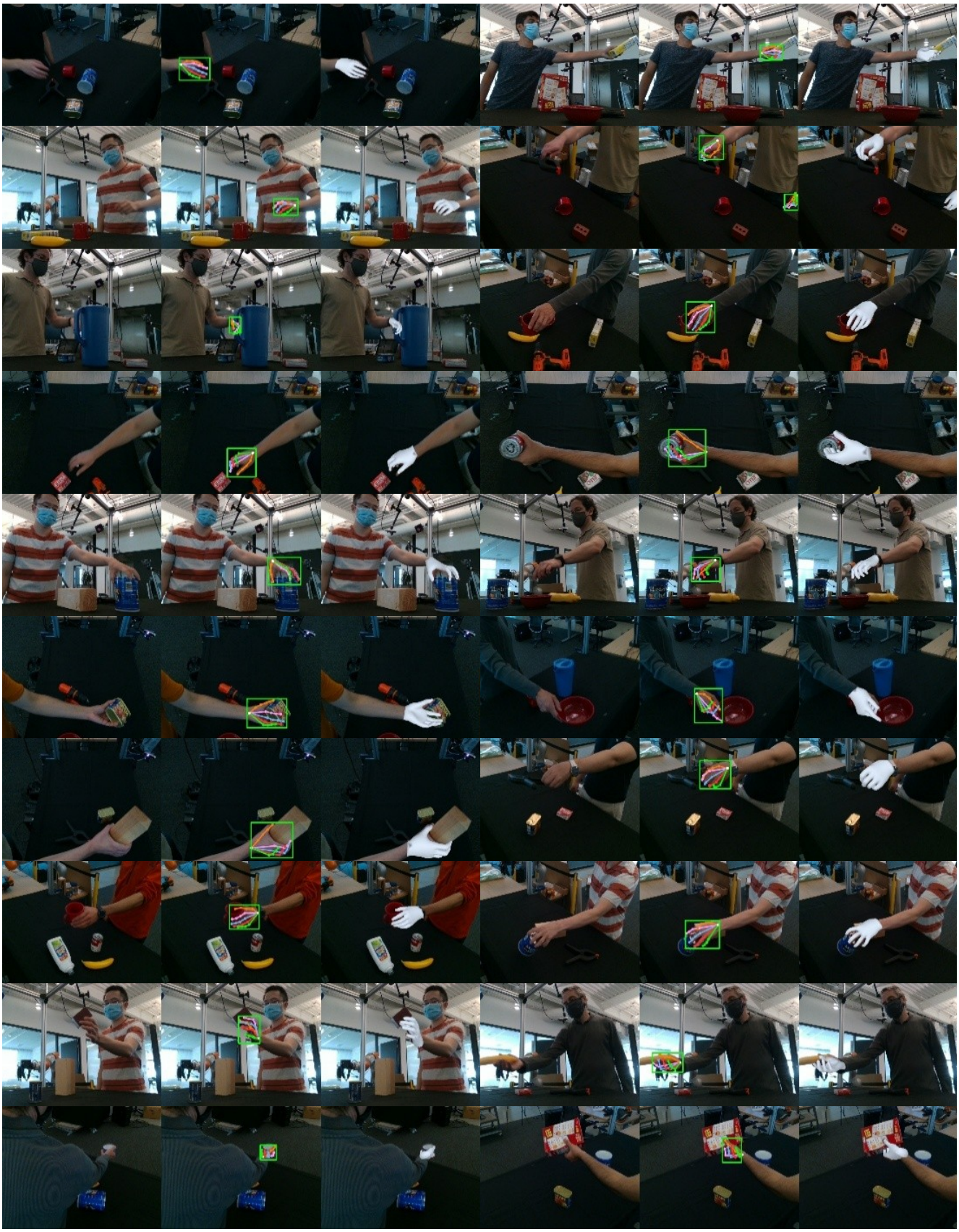

Figure XIV. Visualization of DexYCB test set. Samples in a triplet are input, 2D detection and joints, and 3D mesh.

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

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