

# HORT: Monocular Hand-held Objects Reconstruction with Transformers

Zerui Chen<sup>1</sup> Rolandos Alexandros Potamias<sup>2</sup> Shizhe Chen<sup>1</sup> Cordelia Schmid<sup>1</sup>

<sup>1</sup>Inria, École normale supérieure, CNRS, PSL Research University <sup>2</sup>Imperial College London

firstname.lastname@inria.fr r.potamias@imperial.ac.uk

<https://zerchen.github.io/projects/hort.html>

## Abstract

Reconstructing hand-held objects in 3D from monocular images remains a significant challenge in computer vision. Most existing approaches rely on implicit 3D representations, which produce overly smooth reconstructions and are time-consuming to generate explicit 3D shapes. While more recent methods directly reconstruct point clouds with diffusion models, the multi-step denoising makes high-resolution reconstruction inefficient. To address these limitations, we propose a transformer-based model to efficiently reconstruct dense 3D point clouds of hand-held objects. Our method follows a coarse-to-fine strategy, first generating a sparse point cloud from the image and progressively refining it into a dense representation using pixel-aligned image features. To enhance reconstruction accuracy, we integrate image features with 3D hand geometry to jointly predict the object point cloud and its pose relative to the hand. Our model is trained end-to-end for optimal performance. Experimental results on both synthetic and real datasets demonstrate that our method achieves state-of-the-art accuracy with much faster inference speed, while generalizing well to in-the-wild images.

## 1. Introduction

Reconstructing hand-held objects in 3D from monocular images has shown great potential for widespread real-world applications, ranging from action recognition [4, 21, 23, 26], human-computer interaction [10, 19, 20] to robotics manipulation [17, 53, 72]. Despite notable advancements in this task, achieving robust and efficient 3D reconstruction remains a significant challenge.

Existing approaches for hand-held object reconstruction can be broadly categorized into two types. The first category employs optimization-based algorithms to predict 3D object poses [7, 28, 33, 92, 101] alone or jointly with 3D shapes [24, 36, 96]. While these methods achieve promising reconstruction accuracy, they often rely on additional

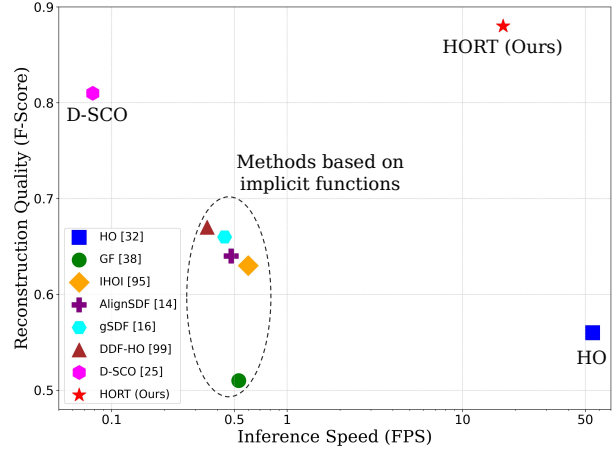


Figure 1. Comparison of inference speed and reconstruction quality on the ObMan dataset. Inference time includes meshing time. Implicit methods [14, 16, 38, 95] require approximately 2 seconds to generate the mesh but fail to achieve compelling results. Explicit methods face trade-offs between reconstruction quality and inference speed: HO [32] uses meshes while D-SCO [25] employs point clouds, but neither achieves optimal performance in both metrics. HORT demonstrates superior performance by achieving high-quality reconstruction with fast inference speed.

3D inputs such as multi-view images, videos, depths or known object meshes. Moreover, they often require several hours to optimize the pose and shape using complex objective functions, making them impractical for real-time applications. To alleviate these issues, the second category explores learning-based approaches. A predominate line of work leverages deep implicit functions [1, 12, 57, 66], such as signed distance functions [14, 16, 38, 71, 95] and directed distance functions [99], to reconstruct hand-held objects. However, methods based on implicit functions tend to generate overly smooth 3D surfaces, failing to capture fine geometric details [18, 66]. Additionally, obtaining explicit 3D meshes from learned implicit fields requires another post-processing step with the Marching Cubes algorithm [51], which not only reduces inference speed (Figure 1) but also limits flexibility for downstream tasks.

A few learning-based methods [25, 32] have explored

explicit 3D representations for hand-held object reconstruction, offering advantages for visualization and integration with downstream applications such as object CAD model registration and optimization-based pipeline [7, 33, 89, 101]. HO [32] employs vertex-based representation, achieving fast inference but suffering from limited reconstruction resolution. Motivated by the recent success of point clouds in 3D reconstruction and generation [40, 55, 63], the state-of-the-art method D-SCO [25] reconstructs high-resolution object point clouds from monocular images using diffusion models [77]. However, despite its strong performance, D-SCO is even slower than implicit function based methods, requiring over 13 seconds per reconstruction due to its intensive denoising steps at test time.

To balance inference speed and 3D reconstruction quality, we introduce HORT, a coarse-to-fine Transformer-based framework for Hand-held Object Reconstruction. HORT consists of four key components: an image encoder, a hand encoder, a sparse point cloud decoder, and a dense point cloud decoder. The two encoders extract complementary features - visual cues from the monocular image and geometric information from the hand. To effectively encode the hand’s kinematic structure, we transform the estimated hand vertices into multiple hand joint coordinate systems before processing them with a PointNet [70] architecture. As hand pose and shape provide implicit cues about the object’s geometry and position, our sparse point cloud decoder integrates both image and 3D hand features via a multi-layer transformer, and jointly predicts sparse object point clouds and hand-relative object pose. The dense point cloud decoder then upsamples these sparse representations to high resolution. For enhanced reconstruction fidelity, we project sparse points onto the image plane to retrieve pixel-aligned image features before applying the dense network for refinement. Our model is trained end-to-end with object reconstruction losses. We evaluate our model on both synthetic ObMan [32] dataset and real-world HO3D [28], DexYCB [9] and MOW [7] datasets. Extensive ablation studies validate the effectiveness of our approach. HORT achieves state-of-the-art reconstruction accuracy across all benchmarks while maintaining high inference speed.

Our contributions are summarized as follows:

- We propose a coarse-to-fine transformer-based framework (HORT) that efficiently reconstructs dense 3D point clouds of hand-held objects from monocular images, balancing reconstruction quality with computational efficiency.
- We integrate fine-grained image and hand geometry features for the sparse and dense point cloud generation, leveraging the implicit cues provided by hand shape for more accurate object reconstruction.
- Our approach outperforms state-of-the-art methods on four diverse benchmarks and generalizes well to in-the-wild images, while achieving fast inference speed.

## 2. Related Work

**3D hand reconstruction.** We have witnessed remarkable progress in 3D hand reconstruction over the past decades, evolving from early efforts in the 1990s [34, 73] to modern deep learning approaches [44, 98]. Early deep learning based approaches [11, 13, 15, 37, 56, 60, 61, 78–80, 82, 91, 102] focus on predicting a sparse set of 3D hand joints from monocular images. The introduction of the parametric MANO hand model [74] has enabled more recent works [2, 5, 11, 30, 42, 43, 46, 47, 52, 62, 67, 68, 75, 85] to reconstruct dense 3D hand mesh with neural networks by integrating MANO as a differentiable layer. More recently, state-of-the-art 3D hand reconstruction methods [67, 68, 100] have shifted from convolutional neural networks to transformer networks [84] and benefitted much from large-scale training, achieving robust 3D hand reconstruction and strong generalization to in-the-wild images. Our work focuses on hand-held object reconstruction, which is more challenging than hands due to the diverse nature of objects. We leverage high-quality automatic hand reconstruction as guidance to improve the accuracy of object reconstruction.

**Hand-held object reconstruction via optimization.** The optimization-based method has been widely used to construct ground-truth 3D reconstructions of hand-held objects [3, 9, 28, 64, 86–88, 94], typically leveraging extra 3D inputs such as multi-view images, depth sensors, or known 3D mesh models to produce accurate reconstructions. However, such inputs are typically unavailable in real-world applications. To address this limitation, another line of work focuses on using optimization to refine initial predictions of object poses and shapes [7, 24, 31, 33, 36, 49, 92, 96, 101]. These methods incorporate constraints to improve hand-object contact, prevent penetration, or enhance temporal consistency in monocular RGB videos. Some methods [7, 33, 92] optimize only 6D object poses given known CAD models with multiple objectives in 2D projection, hand-object interaction and collision. Other works [24, 31, 96] further relax the CAD assumption by jointly optimizing object poses and shapes parameterized by SDF or NeRF [58]. A recent work EasyHOI [49] employs foundation models to reconstruct objects, followed by hand pose optimization to infer plausible hand and object configurations. It demonstrates impressive zero-shot performance on main benchmarks. In general, optimization-based methods require complex objective functions and iterative refinement, making them prone to local minima and leading to slow inference speed. In contrast, our method reconstructs hand-held objects in an efficient feed-forward manner. Our prediction results can also serve as strong initializations for accelerating optimization-based methods.

**Hand-held object reconstruction via learning.** Learning-based methods [14, 16, 25, 32, 35, 45, 83, 95, 99] train neural models to directly reconstruct hand-held objects from

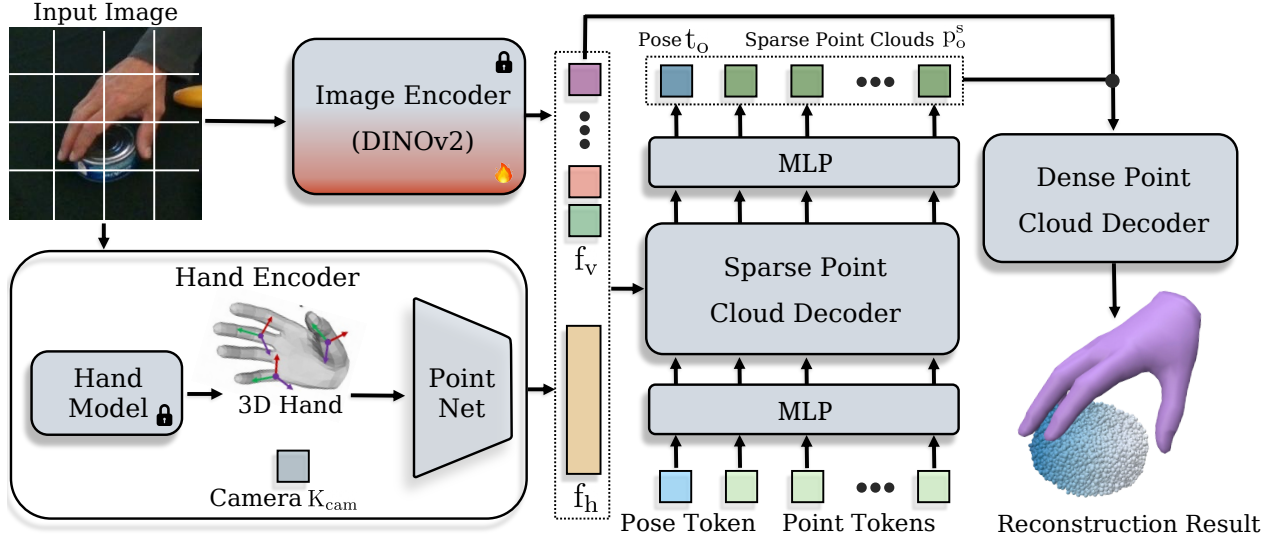


Figure 2. Overview of the proposed HORT model for 3D reconstruction of hand-held objects from a single RGB image. The model first extracts fine-grained visual and hand geometry features using an image encoder and a hand encoder. Then, the sparse and dense point cloud decoders integrate both visual and hand information to progressively generate object point clouds in a coarse-to-fine manner.

monocular images. Early works such as Hasson *et al.* [32] and Tse *et al.* [83] employ AtlasNet [27] to reconstruct objects as sets of mesh vertices, which suffer from limited resolution and unrealistic geometry. To improve detailed reconstruction, later works [14, 16, 38, 95, 99] adopt implicit 3D representations such as SDF. Despite the potential of continuous functions, these methods often generate overly smooth surfaces and are time-consuming to derive explicit 3D shapes. To tackle this issue, the state-of-the-art method D-SCO [96] proposes to reconstruct explicit point clouds with diffusion models. However, the multi-step denoising process in diffusion models leads to low inference speed for high-resolution point clouds. In this work, we adopt explicit point clouds similar to D-SCO. However, our approach differs fundamentally by using a transformer-based architecture to reconstruct object point clouds in a coarse-to-fine manner, achieving state-of-the-art reconstruction accuracy while significantly improving inference speed.

### 3. Method

In this section, we present the **HORT** model for **H**and-held **O**bject **R**econstruction based on **T**ransformers, which predicts dense object point clouds from monocular RGB images. We first introduce the overall architecture in Section 3.1, followed by a description of the detailed components in Section 3.2 to 3.4. Finally, we describe the training objectives for end-to-end optimization of the HORT model in Section 3.5.

#### 3.1. Overview

Figure 2 illustrates the overview of our hand-held object reconstruction model HORT, which consists of four key modules: the image encoder, the hand encoder, the sparse point

cloud decoder and the dense point cloud decoder. We adopt a coarse-to-fine paradigm to progressively generate high-resolution point clouds for hand-held objects.

In the first stage, a sparse point cloud is generated using the image and hand encoders, along with the sparse point cloud decoder. The two encoders extract image features and 3D hand features, respectively. The image features provide crucial information about the object’s appearance and 2D contours, while the 3D hand features help regularize the object’s 3D shape and its spatial relationship to the hand, which enable to mitigate the inherent ambiguities in monocular RGB-based 3D reconstruction. Unlike previous approaches [14, 16, 95] that only leverage sparse hand joint poses for object shape learning, our hand encoder utilizes fine-grained hand geometry, providing richer structural cues to infer the shape of manipulated objects. To effectively integrate the image and hand features, we introduce the sparse point cloud decoder, which jointly predicts the initial sparse point cloud and the object’s pose relative to the hand.

In the second stage, we propose a dense point cloud decoder to refine and upsample the sparse point clouds generated in the first stage. Leveraging the predicted camera parameters, we project the sparse point clouds onto the image plane, enabling to retrieve fine-grained, pixel-aligned image features for each sparse point. These features provide rich local details to enhance the upsampling process, leading to high-resolution 3D object reconstruction. This two-stage framework strikes a balance between reconstruction accuracy and computational efficiency.

#### 3.2. Image and hand feature encoding

**Fine-grained hand features.** Given a single monocular image as the input, we use a state-of-the-art hand pose esti-

mation model [32, 68] to reconstruct the 3D hand and predict the camera parameters  $K_{\text{cam}}$ , which includes a camera translation  $t_c \in \mathbb{R}^3$  and a scale factor  $s_c \in \mathbb{R}$  that normalizes the hand size in the camera coordinate system. The estimated 3D hand mesh is parameterized using MANO [74] and consists of 3D vertices  $v_h \in \mathbb{R}^{778 \times 3}$  driven by rigged hand joints  $j_h \in \mathbb{R}^{16 \times 3}$ . To effectively encode the 3D hand, we transform  $v_h$  into multiple local coordinate systems including all hand joints, finger tips and palm, resulting in a total of 22 coordinate systems (16 joints, 5 fingertips, and 1 palm). We further concatenate the transformed coordinates of each vertex with an absolute vertex index, which helps to better capture the spatial positioning of each vertex within the hand shape. This yields a comprehensive hand representation  $e_h \in \mathbb{R}^{778 \times (22 \times 3 + 1)}$  that captures both pose and shape dependent geometry. We finally encode this rich geometric representation  $e_h$  through an efficient PointNet architecture [70] to extract hand features  $f_h \in \mathbb{R}^{1024}$ .

**Image features.** We employ DINOv2 [65] to extract image features from the input image  $I \in \mathbb{R}^{224 \times 224 \times 3}$ . DINOv2 embeds images into 257 visual feature tokens  $f_v \in \mathbb{R}^{257 \times 1024}$  with a patch size of 14. To balance DINOv2’s pre-trained knowledge from large-scale datasets with domain-specific adaptation for hand-object interactions, we freeze the earlier layers in DINOv2 and only fine-tune its last twelve transformer layers. We leverage both the hand feature  $f_h$  and the image feature  $f_v$  for the following hand-held object reconstruction.

### 3.3. Sparse point cloud reconstruction

We reconstruct the hand-held object under the hand palm coordinate system, as the hand palm can be reliably localized by hand pose estimation models [67, 68]. To achieve this, we decompose the reconstruction task into two sub-tasks: canonical object point cloud generation and hand-relative object pose estimation. This disentanglement separates object shape recovery from pose estimation, facilitating the learning process. For object pose estimation, we predict only the object’s 3D translation relative to the hand palm, denoted as  $t_o \in \mathbb{R}^3$ . This is because many everyday objects exhibit high degrees of symmetry, making rotation prediction an ill-posed problem, as shown in previous works [16, 25]. For sparse point cloud generation, we then produce the sparse object point cloud  $p_o^s \in \mathbb{R}^{N_p^s \times 3}$  in its rotated state, where  $N_p^s$  denotes the number of sparse points.

The sparse point cloud decoder employs a unified transformer backbone to jointly predict the object pose  $t_o$  and the rotated point cloud  $p_o^s$ . Specifically, we define  $1 + N_p^s$  learnable token embeddings: one token for translation prediction and  $N_p^s$  tokens for point cloud generation. These tokens are processed by a multi-layer transformer model, which first applies self-attention among the tokens, followed by cross-attention to condition on the extracted image

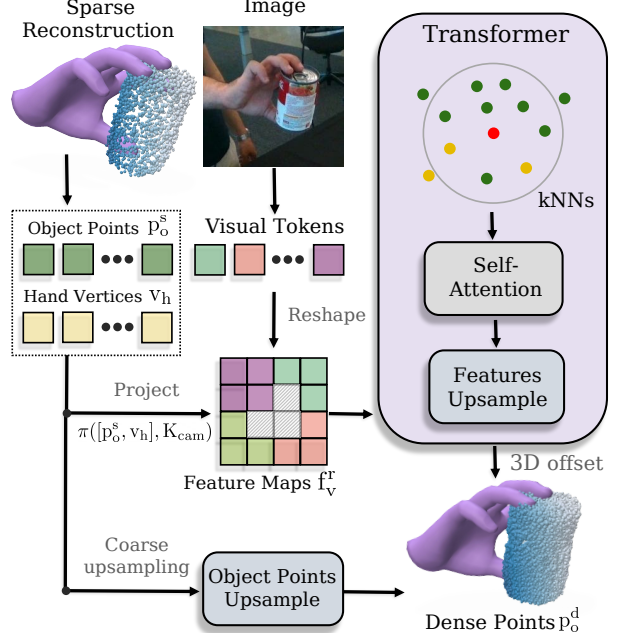


Figure 3. Illustration of our dense point cloud decoder. The green and yellow points in the transformer indicate object points and hand vertices respectively. The model retrieves pixel-aligned image features for each reconstructed point and enhances its local context through self-attention. It upsamples the sparse point cloud to a high-resolution 3D object point cloud.

features  $f_v$  and hand features  $f_h$ . The attention mechanism allows effective feature fusion across different modalities. The output embedding of the first token is used to predict hand-relative object translation  $t_o$ , while the remaining  $N_p^s$  output embeddings generate point cloud coordinates  $p_o^s$ . By sharing a single transformer backbone for predicting object pose and point cloud, the decoder effectively leverages mutual dependencies between the two tasks and is more efficient compared to prior work [25] that trains a separate object pose estimation network.

### 3.4. Dense point cloud reconstruction

Given the predicted sparse point clouds  $p_o^s$  and the hand-relative object pose  $t_o$ , the dense point cloud decoder further upscales it to produce high-resolution point cloud of the hand-held object. Figure 3 provides details of the proposed dense point cloud decoder.

We first bilinearly upsample the reconstructed point clouds  $p_o^s$ . However, the bilinearly upscaled points only provide a coarse approximation of the dense geometry, lacking alignment with image features and local contextual neighborhood information for each point. Therefore, we introduce an additional pixel-aligned feature extraction method that provides localized visual information to the dense point clouds decoder. Specifically, we first exclude the global [CLS] token from  $f_v \in \mathbb{R}^{257 \times 1024}$  to focus only on the patch-level tokens. These remaining tokens are then

reshaped into a spatial grid format and processed through  $3 \times 3$  convolution layers, yielding a refined feature map  $f_v^r \in \mathbb{R}^{16 \times 16 \times 128}$ . Next, we transform  $p_o^s$  from the object canonical space to the camera coordinate space using the estimated hand palm location  $t_p$  and hand-relative object translation  $t_o$ . Using the predicted camera parameters  $K_{cam}$ , we project  $p_o^s$  onto the image plane and retrieve pixel-aligned features for each point via perspective projection  $\pi(\cdot)$  and bilinear interpolation  $F(\cdot)$ :

$$f_o = F(\pi(p_o^s + t_p + t_o, K_{cam}), f_v^r). \quad (1)$$

The coordinates in the sparse point cloud  $p_o^s \in \mathbb{R}^{N_p^s \times 3}$  are concatenated with their aligned visual features  $f_o \in \mathbb{R}^{N_p^s \times 128}$ . Similarly, we also retrieve visual features for 3D hand vertices  $v_h$  and feed them into the transformer model with object features. The model consists of two blocks, each performing self-attention followed by feature upsampling. The point cloud resolution is increased by factors of 2 and 4 in the first and second blocks, respectively. Specifically, as shown in Figure 3, each block first applies self-attention [84] within local neighborhoods to aggregate spatial and visual hand-object contexts for each object point, inspired by [90]. The neighborhoods are determined using  $k$  Nearest Neighbors (kNNs) based on spatial locations of the points, with  $k = 16$  set empirically. The point features are then bilinearly upsampled and passed through three convolutional layers to predict 3D offsets for each upsampled point. This process ultimately reconstructs a high-resolution dense point cloud  $p_o^d \in \mathbb{R}^{N_p^d \times 3}$ .

### 3.5. Training

Unlike D-SCO [25] that trains each component separately, we adopt an end-to-end training strategy to optimize the entire HORT model. All components are jointly trained using an overall loss function  $\mathcal{L}$  combining object pose loss along with sparse and dense point cloud generation losses:

$$\mathcal{L} = \lambda_1 \times \mathcal{L}_{pose} + \lambda_2 \times \mathcal{L}_{cd}^s + \mathcal{L}_{cd}^d. \quad (2)$$

where  $\lambda_1$  and  $\lambda_2$  are hyper-parameters to balance losses.

Specifically, we use the  $\ell_1$  loss  $\mathcal{L}_{pose}$  to predict the 3D object translation  $t_o$  relative to the hand palm as follows:

$$\mathcal{L}_{pose} = \left\| t_o - \hat{t}_o \right\|_1, \quad (3)$$

where  $\hat{t}_o$  denotes the ground-truth 3D offset between the object centroid and the palm.

We employ the Chamfer Distance loss  $\mathcal{L}_{cd}^s$  to supervise 3D point clouds generation as follows:

$$\mathcal{L}_{cd}^{\{s,d\}} = \text{Chamfer Distance}(p_o^{\{s,d\}}, \hat{p}_o^{\{s,d\}}), \quad (4)$$

where  $p_o^{\{\cdot\}}$  denote either sparse ( $p_o^s$ ) or dense ( $p_o^d$ ) object point clouds.  $\hat{p}_o^{\{\cdot\}}$  represent their counterparts evenly sampled from the ground-truth object mesh.

## 4. Experiments

We conduct comprehensive experiments to validate the effectiveness of our approach on four diverse hand-held object reconstruction benchmarks.

### 4.1. Datasets

**ObMan** [32]: It is a large-scale synthetic hand-object interaction dataset, containing 2,772 object instances from ShapeNet [8] with 21K hand grasps generated by GraspIt! [59]. For fair comparison with previous works, we use the standard split which includes 141K and 6,285 samples for training and testing respectively. Following [14], we randomly sample 30K samples from the standard training set for ablation experiments, while the full training data is used for comparison with state-of-the-art methods.

**HO3D** [28]: We use the HO3Dv3 version [29] of this dataset, which captures 68 videos sequences of ten subjects manipulating ten YCB objects [6]. We follow the standard experimental setup in [25, 95, 99] and include 77K samples for training and 1,221 samples for evaluation.

**DexYCB** [9]: It captures hand grasping of twenty YCB objects from eight cameras by twenty subjects. Following the convention in [14, 16, 93], we focus on the right-hand samples and use the official s0 split. We follow the gSDF [16] setup to downsample the video every 5 frames, resulting in 29K training images and 5,928 testing samples.

**MOW** [7]: It contains 520 images sampled from YouTube videos in 100 Days of Hands dataset [76]. The training and testing split contain 350 and 92 images respectively.

### 4.2. Evaluation metrics

Following prior works [16, 32, 95], we use two main metrics to evaluate the 3D reconstruction quality. We also report interaction metrics to provide more details about the plausibility of our reconstructed hand-object configurations.

**Chamfer Distance (CD)**. We compute Chamfer Distance ( $\text{cm}^2$ ) between the reconstructed dense object point clouds and the ground-truth object point clouds. Following the standard protocol from previous works [16], we sample 30K points on both point clouds to compute Chamfer distance. We report the mean Chamfer distance over the whole test set to evaluate the quality of our reconstructed point clouds.

**F-score (FS)**. Since Chamfer distance is sensitive to outliers [81, 95], we also report the F-score, a geometric mean of precision and recall, to comprehensively evaluate the reconstructed object point clouds and report F-score at 5mm (FS@5) and 10mm (FS@10) thresholds.

**Contact Ratio (Cr)**. We follow previous works [14, 16, 38] to report the percentage of testing samples where contact points between the reconstructed hand and object exist.

**Penetration Depth (Pd)**. Following [14, 16, 38], we compute the maximum distance (in cm) from the estimated hand mesh vertices to the object’s surface in case of a collision.

Table 1. The performance of our hand-held object reconstruction models with variants of visual feature encoders on ObMan dataset.

|    | Hand Encoder |        | Image Encoder | FS@5 ↑      | FS@10 ↑     | CD ↓       | Cr ↑        | Pd ↓        |
|----|--------------|--------|---------------|-------------|-------------|------------|-------------|-------------|
|    | Palm         | Joints |               |             |             |            |             |             |
| R1 | ×            | ×      | Fine-tune     | 0.45        | 0.75        | 3.1        | 0.76        | 2.15        |
| R2 | ✓            | ×      | Fine-tune     | 0.53        | 0.82        | 2.4        | 0.85        | 1.76        |
| R3 | ✓            | ✓      | Fine-tune     | <b>0.60</b> | <b>0.87</b> | <b>1.8</b> | <b>0.92</b> | <b>1.37</b> |
| R4 | ✓            | ✓      | Scratch       | 0.51        | 0.79        | 2.6        | 0.88        | 1.48        |
| R5 | ✓            | ✓      | Frozen        | 0.48        | 0.76        | 2.9        | 0.87        | 1.62        |

Table 2. The performance of our hand-held object reconstruction models with various object decoder variants on ObMan dataset. J-decode indicates models that jointly decode sparse point clouds with hand-relative 3D translation using a transformer architecture.

|    | Sparse   |          | Dense         |            | FS@5 ↑      | FS@10 ↑     | CD ↓       |
|----|----------|----------|---------------|------------|-------------|-------------|------------|
|    | J-decode | upsample | aligned feat. | hand feat. |             |             |            |
| R1 | ×        | ×        | ×             | ×          | 0.41        | 0.69        | 3.8        |
| R2 | ✓        | ×        | ×             | ×          | 0.47        | 0.77        | 3.0        |
| R3 | ✓        | ✓        | ×             | ×          | 0.54        | 0.82        | 2.3        |
| R4 | ✓        | ✓        | ✓             | ×          | 0.58        | 0.85        | 2.0        |
| R5 | ✓        | ✓        | ✓             | ✓          | <b>0.60</b> | <b>0.87</b> | <b>1.8</b> |

### 4.3. Implementation details

**Model architecture.** We use the large DINOv2 model [65] as our image encoder, which consists of 24 transformer layers. The PointNet [70] for hand feature extraction contains 5 MLP layers. The sparse point cloud decoder consists of 10 transformer layers with 8 attention heads. The learnable token for each point has 512 dimensional features. The initially decoded sparse object point clouds contains  $N_p^s = 2,048$  points. Followed by 3 layers of point deconvolution and transformer layers, we finally reconstruct a dense point cloud of  $N_p^d = 16,384$  object points, which is the same number of points used in D-SCO [25].

**Training details.** We crop the hand-object region out of the original image like previous works [14, 32, 95]. In training, according to [54, 97], we adjust camera intrinsics and extrinsics according to the bounding box location and resizing ratio. The input image size for our model is  $224 \times 224$ . We perform data augmentation including random rotation ( $[-90^\circ, 90^\circ]$ ), random resizing ( $[0.8, 1.2]$ ), random bounding box shifting and color jittering. During training, we use ground-truth camera parameters and hand poses, but our approach does not assume that they are available at test time. We train our models with a batch size of 192 for 50 epochs on both ObMan and DexYCB datasets. We use the Adam optimizer [41] with a base learning rate of  $1e-4$  and the cosine learning rate decay. We empirically set both  $\lambda_1$  and  $\lambda_2$  to two. We fine-tune our ObMan pre-trained model on HO3D, DexYCB and MOW datasets for a fair comparison with previous works. The training takes around 50 hours on ObMan dataset with 4 NVIDIA A100 GPUs.

### 4.4. Ablation studies

We conduct extensive ablation studies on the ObMan dataset to evaluate the effectiveness of each component in

Table 3. Comparison with state-of-the-art methods on ObMan dataset. HORT achieves superior performance across all metrics when using either predicted or ground-truth 3D hands.

| Methods       | Predicted hand |             |            | Ground-truth hand |             |            |
|---------------|----------------|-------------|------------|-------------------|-------------|------------|
|               | FS@5 ↑         | FS@10 ↑     | CD ↓       | FS@5 ↑            | FS@10 ↑     | CD ↓       |
| HO [32]       | 0.23           | 0.56        | 6.4        | -                 | -           | -          |
| GF [38]       | 0.30           | 0.51        | 13.9       | -                 | -           | -          |
| IHOI [95]     | 0.42           | 0.63        | 10.2       | 0.49              | 0.70        | 9.2        |
| AlignSDF [14] | 0.40           | 0.64        | 9.2        | -                 | -           | -          |
| gSDF [16]     | 0.44           | 0.66        | 8.8        | -                 | -           | -          |
| DDF-HO [99]   | 0.55           | 0.67        | 1.4        | -                 | -           | -          |
| D-SCO [25]    | 0.61           | 0.81        | 1.1        | 0.65              | 0.85        | 1.0        |
| HORT (Ours)   | <b>0.66</b>    | <b>0.88</b> | <b>1.0</b> | <b>0.72</b>       | <b>0.91</b> | <b>0.9</b> |

Table 4. Comparison with state-of-the-art methods on HO3D and DexYCB. HORT achieves the best performance on both datasets.

| Methods       | HO3D        |             |            | DexYCB      |             |            |
|---------------|-------------|-------------|------------|-------------|-------------|------------|
|               | FS@5 ↑      | FS@10 ↑     | CD ↓       | FS@5 ↑      | FS@10 ↑     | CD ↓       |
| HO [32]       | 0.11        | 0.22        | 41.9       | 0.38        | 0.64        | 4.2        |
| GF [38]       | 0.12        | 0.24        | 49.6       | 0.39        | 0.66        | 4.5        |
| IHOI [95]     | 0.28        | 0.50        | 15.3       | -           | -           | -          |
| AlignSDF [14] | -           | -           | -          | 0.41        | 0.68        | 3.9        |
| gSDF [16]     | -           | -           | -          | 0.44        | 0.71        | 3.4        |
| DDF-HO [99]   | 0.27        | 0.40        | 8.6        | -           | -           | -          |
| D-SCO [25]    | 0.41        | 0.63        | 3.4        | <b>0.63</b> | 0.82        | 1.3        |
| HORT (Ours)   | <b>0.45</b> | <b>0.72</b> | <b>2.6</b> | <b>0.63</b> | <b>0.85</b> | <b>1.1</b> |

our model. We follow [14] to use randomly sampled 30K data for training and the original test set for evaluation.

**Hand encoder.** The first three rows in Table 1 evaluate the contributions of our proposed hand encoder. The base model in R1, which relies solely on image features without a hand encoder, achieves poor results. R2 incorporates the 3D hand vertices under the hand palm coordinate system as input, which significantly improves performance across all metrics, for example, the FS@5 improves from 0.45 to 0.53 and CD from 3.1 to 2.4. In addition, it also greatly improves the contact ratio Cr and reduces the penetration depth Pd. This confirms that hand information substantially enhances reconstruction accuracy of hand-held objects and better recover interaction between the hand and the object. Further enhancement of hand features in R3, where hand vertices are transformed into multiple hand joint coordinate systems, yields additional gains particularly for FS@5 increased from 0.53 to 0.60. These results clearly demonstrate the necessity of effectively encoding hand geometry for accurate hand-held object reconstruction.

**Image encoder.** In the last three rows of Table 1, we compare different training strategies for the image encoder. R4 trains a ViT image encoder [89] from scratch on the ObMan dataset, while R5 uses the frozen DINOv2 model. Both approaches perform worse than our fine-tuning strategy in R3 which refines the visual features for the hand-object interaction domain.

**Sparse point cloud decoder.** We evaluate our joint decoding strategy for the sparse point cloud decoder in the first two rows of Table 2. The model in R1 employs two separate modules: one for object pose prediction, which con-



Figure 4. Qualitative results of the proposed model on diverse benchmarks. The first row shows our results on the synthetic ObMan dataset. The second, third and forth rows present our results on HO3D, DexYCB and MOW datasets respectively. The fifth row illustrates our performance on in-the-wild CORE50 images [50]. The last row shows the results of our model on images captured by our mobile phone. Our approach can accurately reconstruct point clouds of hand-held objects from both synthetic and real-world images.

catenates the global image feature (*i.e.*, [CLS] token in DI-NOv2) with 3D hand features from PointNet as input to an MLP for translation prediction; and another for point cloud generation similar to our final sparse decoder. In contrast, the model in R2 jointly predicts the object pose and shape through a unified transformer decoder. This design integrates object volume features for more accurate object localization, leading to enhanced performance across metrics.

**Dense point cloud decoder.** We further evaluate our coarse-to-fine strategy and dense point cloud decoder design in the last four rows of Table 2. In R3, we concatenate coordinates of the sparse point cloud with global image features before feeding them to the dense decoder. Compared to R2, which lacks the point cloud upsampling step, R3 shows significant improvement in reconstruction performance. R4 further replaces the global image features used in R3 with pixel-aligned features and improves the performance. As shown in Figure 3, R5 additionally includes hand features and integrate both hand and object contexts in the transformer model for object points upsampling, which achieves the best results across all evaluation metrics.

#### 4.5. Comparison with state of the art

We compare our approach with state-of-the-art methods on four benchmarks. Figure 4 presents some qualitative examples of our HORT model on these datasets.

**ObMan.** Table 3 presents a comprehensive comparison on ObMan dataset, where we evaluate models under two testing scenarios: using either ground-truth or estimated hand poses as input. For the estimated hand poses, we follow the same practice as in previous works [25, 95] to train a hand pose estimation model [32] for ObMan. Our approach consistently outperforms state-of-the-art methods under both settings. Our model shows substantial improvement over methods based on implicit functions (*i.e.*, DDF-HO [99] and gSDF [16]) and the recent D-SCO [25] method.

**HO3D.** In the left block of Table 4, we compare different methods on the real-world HO3D benchmark. We use predicted hand poses and camera parameters from an off-the-shelf hand model [68] and fine-tune the model from the ObMan pre-trained model for fair comparison with prior works [25, 95, 99]. Our model constantly outperforms previous methods under all evaluation metrics. Compared with

Table 5. Comparison with state-of-the-art methods on MOW. HORT achieves the best performance under both evaluation setup.

| Methods      | Fine-tune from ObMan |                  |                 | Multiple datasets |                  |                 |
|--------------|----------------------|------------------|-----------------|-------------------|------------------|-----------------|
|              | FS@5 $\uparrow$      | FS@10 $\uparrow$ | CD $\downarrow$ | FS@5 $\uparrow$   | FS@10 $\uparrow$ | CD $\downarrow$ |
| IHOI [95]    | 0.13                 | 0.24             | 49.8            | -                 | -                | -               |
| DDF-HO [99]  | 0.17                 | 0.24             | 15.9            | -                 | -                | -               |
| D-SCO [25]   | <b>0.30</b>          | 0.47             | 13.5            | -                 | -                | -               |
| WildHOI [69] | -                    | -                | -               | 0.08              | 0.20             | 27.2            |
| MCC-HO [89]  | -                    | -                | -               | 0.15              | 0.31             | 15.2            |
| HORT (Ours)  | <b>0.30</b>          | <b>0.49</b>      | <b>10.2</b>     | <b>0.37</b>       | <b>0.57</b>      | <b>5.4</b>      |

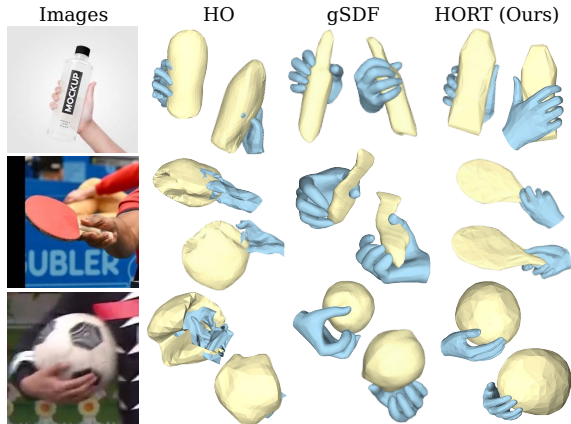


Figure 5. Qualitative comparison with previous methods [16, 32]. D-SCO [25], our approach achieves relative improvement of 14.3% and 23.5% for FS@10 and CD respectively.

**DexYCB.** The right block of Table 4 presents our experimental results on the DexYCB dataset. We follow prior works [25, 99] to use a hand model [71] that has been specifically trained on this dataset. As shown in Table 4, our model outperforms the strong video-based gSDF model [16] and the state-of-the-art D-SCO method [25], achieving better FS@10 and CD.

**MOW.** We conduct experiments on the MOW dataset under two settings for fair comparison with previous works. In the first setting [25, 95, 99], we pre-train the model on ObMan and fine-tune it on MOW. In the second setting [89], we train our model jointly on the DexYCB, HOI4D [48] and MOW datasets. As shown in Table 5, our model outperforms previous methods in both settings, demonstrating the robustness of our method on challenging YouTube samples.

**Generalization to in-the-wild images.** Figure 4 shows the performance of our model on CORE50 images [50] (5<sub>th</sub> row) and images captured by our phone (6<sub>th</sub> row). It demonstrates that our model generalizes robustly to diverse objects and textures after joint training on the DexYCB, HOI4D, and MOW datasets, as shown in Table 5. We also qualitatively compare HORT with previous approaches [16, 32] in Figure 5, which demonstrates that HORT produces more convincing results. Please see appendix for our robustness to occlusion and failure cases analysis.

**Runtime comparison.** As shown in Table 6, we compare the runtime and computational cost of our method with previous approaches on the ObMan dataset. Inference speed

Table 6. Runtime comparison with previous methods on ObMan. HORT significantly reduces computational cost and accelerates model inference. We report the inference speed of our HORT algorithm both with and without the meshing step.

| Methods       | #points | #verts | GFLOPs $\downarrow$ | Params $\downarrow$ | fps $\uparrow$ | fps mesh $\uparrow$ | CD $\downarrow$ |
|---------------|---------|--------|---------------------|---------------------|----------------|---------------------|-----------------|
| HO [32]       | N/A     | 0.6k   | <b>76</b>           | 25M                 | <b>55.2</b>    | <b>55.2</b>         | 6.4             |
| GF [38]       | N/A     | 20k    | 6.4k                | 26M                 | 0.37           | 0.37                | 13.9            |
| IHOI [95]     | N/A     | 20k    | 5.9k                | <b>24M</b>          | 0.37           | 0.37                | 10.2            |
| AlignSDF [14] | N/A     | 20k    | 6.6k                | 33M                 | 0.48           | 0.48                | 9.2             |
| gSDF [16]     | N/A     | 20k    | 6.7k                | 35M                 | 0.44           | 0.44                | 8.8             |
| DDF-HO [99]   | N/A     | 20k    | 8.9k                | 25M                 | 0.35           | 0.35                | 1.4             |
| D-SCO [25]    | 16k     | 1k     | -                   | -                   | 0.08           | -                   | 1.1             |
| HORT (Ours)   | 16k     | 1k     | 0.25k               | 366M                | 16.9           | 4.8                 | <b>1.0</b>      |

and GFLOPs are measured on an NVIDIA A100 GPU for hand-held object reconstruction including hand prediction. Implicit methods require inference on densely sampled points (*i.e.*,  $128 \times 128 \times 128$  in common practice) at test time, resulting in over 6000 GFLOPs. D-SCO [25] is even slower than implicit models due to intensive denoising steps. HO [32] is very fast but suffers from low-quality 3D reconstruction. Our model requires only 258 GFLOPs and achieves robust performance, running at 16.9 fps including both 3D hand and object reconstruction.

**Point clouds versus meshes.** Given the reconstructed point clouds, fitting meshes using off-the-shelf algorithms [22, 39] is straightforward. This process does not affect performance; *e.g.*, FS@10 remains at 0.85 on the DexYCB dataset. However, it increases the runtime, reducing the speed from 16.9 fps to 4.8 fps. Despite this, our model remains significantly faster than methods based on implicit functions and the recent D-SCO [25] method in Table 6.

## 5. Conclusion

This work presents HORT, a transformer-based framework for reconstructing dense point clouds of hand-held objects from monocular RGB images. Our approach follows an efficient coarse-to-fine learning paradigm to first generate a sparse object point cloud and progressively refine it to a high-fidelity dense point cloud. To further improve reconstruction accuracy, we propose to extract fine-grained 3D hand geometry features together with image information for joint object pose and point clouds prediction, as well as pixel-aligned features for upsampling. Comprehensive experiments across four benchmarks including ObMan, HO3D, DexYCB and MOW consistently demonstrate the superiority of our method over existing approaches.

**Acknowledgement:** This work was granted access to HPC resources of IDRIS under the allocation AD011013147 made by GENCI. It was funded in part by the French government under management of Agence Nationale de la Recherche as part of the “France 2030” program, reference ANR-23-IACL-0008 (PR[AI]RIE-PSAI projet), and the ANR project VideoPredict (ANR-21-FAII-0002-01) and the Paris Île-de-France Région in the frame of the DIM AI4IDF. R.A. Potamias was partially supported by EPSRC Project GNOMON (EP/X011364). Cordelia Schmid would like to acknowledge the support by the Körber European Science Prize.

## References

- [1] Tristan Aumentado-Armstrong, Stavros Tsogkas, Sven Dickinson, and Allan D Jepson. Representing 3D shapes with probabilistic directed distance fields. In *CVPR*, 2022. 1
- [2] 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
- [3] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. In *ECCV*, 2012. 2
- [4] Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Fan Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, et al. Introducing HOT3D: An egocentric dataset for 3D hand and object tracking. *arXiv:2406.09598*, 2024. 1
- [5] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3D hand shape and pose from images in the wild. In *CVPR*, 2019. 2
- [6] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The YCB object and model set: Towards common benchmarks for manipulation research. In *ICAR*, 2015. 5
- [7] Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. Reconstructing hand-object interactions in the wild. In *ICCV*, 2021. 1, 2, 5, 14
- [8] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 5
- [9] 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, 14
- [10] Yu-Wei Chao, Chris Paxton, Yu Xiang, Wei Yang, Balakumar Sundaralingam, Tao Chen, Adithyavairavan Murali, Maya Cakmak, and Dieter Fox. HandoverSim: A simulation framework and benchmark for human-to-robot object handovers. In *ICRA*, 2022. 1
- [11] 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
- [12] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, 2019. 1
- [13] Zerui Chen, Yan Huang, Hongyuan Yu, Bin Xue, Ke Han, Yiru Guo, and Liang Wang. Towards part-aware monocular 3D human pose estimation: An architecture search approach. In *ECCV*, 2020. 2
- [14] Zerui Chen, Yana Hasson, Cordelia Schmid, and Ivan Laptev. AlignSDF: Pose-Aligned signed distance fields for hand-object reconstruction. In *ECCV*, 2022. 1, 2, 3, 5, 6, 8, 13
- [15] Zerui Chen, Yan Huang, Hongyuan Yu, and Liang Wang. Learning a robust part-aware monocular 3d human pose estimator via neural architecture search. *IJCV*, 2022. 2
- [16] Zerui Chen, Shizhe Chen, Cordelia Schmid, and Ivan Laptev. gSDF: Geometry-driven signed distance functions for 3D hand-object reconstruction. In *CVPR*, 2023. 1, 2, 3, 4, 5, 6, 7, 8, 13
- [17] Zerui Chen, Shizhe Chen, Etienne Arlaud, Ivan Laptev, and Cordelia Schmid. ViViDex: Learning vision-based dexterous manipulation from human videos. In *ICRA*, 2025. 1
- [18] Jaesung Choe, Byeongin Joung, Francois Rameau, Jaesik Park, and In So Kweon. Deep point cloud reconstruction. In *ICLR*, 2022. 1
- [19] Sammy Christen, Wei Yang, Claudia Pérez-D’Arpino, Otmar Hilliges, Dieter Fox, and Yu-Wei Chao. Learning human-to-robot handovers from point clouds. In *CVPR*, 2023. 1
- [20] Sammy Christen, Lan Feng, Wei Yang, Yu-Wei Chao, Otmar Hilliges, and Jie Song. SynH2R: Synthesizing hand-object motions for learning human-to-robot handovers. In *ICRA*, 2024. 1
- [21] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The EPIC-KITCHENS dataset. In *ECCV*, 2018. 1
- [22] Herbert Edelsbrunner, David Kirkpatrick, and Raimund Seidel. On the shape of a set of points in the plane. *IEEE Transactions on information theory*, 1983. 8, 13
- [23] Zicong Fan, Takehiko Ohkawa, Linlin Yang, Nie Lin, Zhis-han Zhou, Shihao Zhou, Jiajun Liang, Zhong Gao, Xu-anyang Zhang, Xue Zhang, et al. Benchmarks and challenges in pose estimation for egocentric hand interactions with objects. In *ECCV*, 2024. 1
- [24] Zicong Fan, Maria Parelli, Maria Eleni Kadoglou, Xu Chen, Muhammed Kocabas, Michael J Black, and Otmar Hilliges. HOLD: Category-agnostic 3D reconstruction of interacting hands and objects from video. In *CVPR*, 2024. 1, 2
- [25] Bowen Fu, Gu Wang, Chenyangguang Zhang, Yan Di, Ziqin Huang, Zhiying Leng, Fabian Manhardt, Xiangyang Ji, and Federico Tombari. D-SCo: Dual-stream conditional diffusion for monocular hand-held object reconstruction. In *ECCV*, 2024. 1, 2, 4, 5, 6, 7, 8, 13
- [26] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 1
- [27] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3D surface generation. In *CVPR*, 2018. 3
- [28] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. HOnnotate: A method for 3D annotation of hand and object poses. In *CVPR*, 2020. 1, 2, 5, 13, 14
- [29] Shreyas Hampali, Sayan Deb Sarkar, and Vincent Lepetit. HO3Dv3: Improving the accuracy of hand-object

- annotations of the HO-3D dataset. *arXiv preprint arXiv:2107.00887*, 2021. 5
- [30] 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*, 2022. 2
  - [31] Shreyas Hampali, Tomas Hodan, Luan Tran, Lingni Ma, Cem Keskin, and Vincent Lepetit. In-Hand 3D object scanning from an RGB sequence. In *CVPR*, 2023. 2
  - [32] 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. 1, 2, 3, 4, 5, 6, 7, 8, 14
  - [33] Yana Hasson, Gül Varol, Cordelia Schmid, and Ivan Laptev. Towards unconstrained joint hand-object reconstruction from RGB videos. In *3DV*, 2021. 1, 2
  - [34] Tony Heap and David Hogg. Towards 3D hand tracking using a deformable model. In *FG*, 1996. 2
  - [35] Junxing Hu, Hongwen Zhang, Zerui Chen, Mengcheng Li, Yunlong Wang, Yebin Liu, and Zhenan Sun. Learning explicit contact for implicit reconstruction of hand-held objects from monocular images. In *AAAI*, 2024. 2
  - [36] Di Huang, Xiaopeng Ji, Xingyi He, Jiaming Sun, Tong He, Qing Shuai, Wanli Ouyang, and Xiaowei Zhou. Reconstructing hand-held objects from monocular video. In *SIGGRAPH Asia*, 2022. 1, 2
  - [37] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5D heatmap regression. In *ECCV*, 2018. 2
  - [38] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. Grasping Field: Learning implicit representations for human grasps. In *3DV*, 2020. 1, 3, 5, 6, 8, 13
  - [39] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *SGP*, 2006. 8
  - [40] Donghwan Kim and Tae-Kyun Kim. Multi-hypotheses conditioned point cloud diffusion for 3d human reconstruction from occluded images. *NeurIPS*, 2024. 2
  - [41] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
  - [42] Dominik Kulon, Haoyang Wang, Riza Alp Güler, Michael M. Bronstein, and Stefanos Zafeiriou. Single image 3D hand reconstruction with mesh convolutions. In *BMVC*, 2019. 2
  - [43] Dominik Kulon, Riza Alp Güler, I. Kokkinos, M. Bronstein, and S. Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, 2020. 2
  - [44] Vincent Lepetit. Recent advances in 3D object and hand pose estimation. *arXiv preprint arXiv:2006.05927*, 2020. 2
  - [45] Kailin Li, Lixin Yang, Haoyu Zhen, Zenan Lin, Xinyu Zhan, Licheng Zhong, Jian Xu, Kejian Wu, and Cewu Lu. CHORD: Category-level hand-held object reconstruction via shape deformation. In *ICCV*, 2023. 2
  - [46] 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. 2
  - [47] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021. 2
  - [48] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. HOI4D: A 4D egocentric dataset for category-level human-object interaction. In *CVPR*, 2022. 8
  - [49] Yumeng Liu, Xiaoxiao Long, Zemin Yang, Yuan Liu, Marc Habermann, Christian Theobalt, Yuexin Ma, and Wenping Wang. EasyHOI: Unleashing the power of large models for reconstructing hand-object interactions in the wild. In *CVPR*, 2025. 2
  - [50] Vincenzo Lomonaco and Davide Maltoni. CORE50: a new dataset and benchmark for continuous object recognition. In *CoRL*, 2017. 7, 8, 14
  - [51] William E Lorensen and Harvey E Cline. Marching Cubes: A high resolution 3D surface construction algorithm. *TOG*, 1987. 1
  - [52] Jun Lv, Wenqiang Xu, Lixin Yang, Sucheng Qian, Chongzhao Mao, and Cewu Lu. HandTailor: Towards high-precision monocular 3D hand recovery. In *BMVC*, 2021. 2
  - [53] Priyanka Mandikal and Kristen Grauman. DexVIP: Learning dexterous grasping with human hand pose priors from video. In *CoRL*, 2022. 1
  - [54] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3D human pose estimation in the wild using improved CNN supervision. In *3DV*, 2017. 6
  - [55] Luke Melas-Kyriazi, Christian Rupprecht, and Andrea Vedaldi. PC2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction. In *CVPR*, 2023. 2
  - [56] Hao Meng, Sheng Jin, Wentao Liu, Chen Qian, Mengxiang Lin, Wanli Ouyang, and Ping Luo. 3D interacting hand pose estimation by hand de-occlusion and removal. In *ECCV*, 2022. 2
  - [57] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D reconstruction in function space. In *CVPR*, 2019. 1
  - [58] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2
  - [59] Andrew T Miller and Peter K Allen. Graspit! a versatile simulator for robotic grasping. *RAM*, 2004. 5
  - [60] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map. In *CVPR*, 2018. 2
  - [61] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Generated hands for real-time 3D hand tracking from monocular RGB. In *CVPR*, 2018. 2

- [62] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A Otaduy, Dan Casas, and Christian Theobalt. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. *TOG*, 2019. 2
- [63] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-E: A system for generating 3D point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2
- [64] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *ICCV*, 2011. 2
- [65] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv:2304.07193*, 2023. 4, 6
- [66] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 1
- [67] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. In *CVPR*, 2024. 2, 4
- [68] Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. WiLoR: End-to-end 3D hand localization and reconstruction in-the-wild. In *CVPR*, 2025. 2, 4, 7
- [69] Aditya Prakash, Matthew Chang, et al. 3d reconstruction of objects in hands without real world 3d supervision. In *ECCV*, 2024. 8
- [70] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 2, 4, 6
- [71] Haozhe Qi, Chen Zhao, Mathieu Salzmann, and Alexander Mathis. HOISDF: Constraining 3D hand-object pose estimation with global signed distance fields. In *CVPR*, 2024. 1, 8
- [72] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. DexMV: Imitation learning for dexterous manipulation from human videos. In *ECCV*, 2022. 1
- [73] James M Rehg and Takeo Kanade. Visual tracking of high DOF articulated structures: an application to human hand tracking. In *ECCV*, 1994. 2
- [74] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied Hands: Modeling and capturing hands and bodies together. *TOG*, 2017. 2, 4
- [75] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. FrankMocap: A monocular 3D whole-body pose estimation system via regression and integration. In *ICCV*, 2021. 2
- [76] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020. 5
- [77] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 2
- [78] Adrian Spurr, Aneesh Dahiya, Xi Wang, Xucong Zhang, and Otmar Hilliges. Self-supervised 3D hand pose estimation from monocular RGB via contrastive learning. In *ICCV*, 2021. 2
- [79] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, 2018.
- [80] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3D articulated hand posture. In *CVPR*, 2014. 2
- [81] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3D reconstruction networks learn? In *CVPR*, 2019. 5
- [82] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+O: Unified egocentric recognition of 3D hand-object poses and interactions. In *CVPR*, 2019. 2
- [83] Tze Ho Elden Tse, Kwang In Kim, Ales Leonardis, and Hyung Jin Chang. Collaborative learning for hand and object reconstruction with attention-guided graph convolution. In *CVPR*, 2022. 2, 3
- [84] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 5
- [85] Jiayi Wang, Franziska Mueller, Florian Bernard, Suzanne Sorli, Oleksandr Sotnychenko, Neng Qian, Miguel A Otaduy, Dan Casas, and Christian Theobalt. RGB2Hands: Real-time tracking of 3D hand interactions from monocular RGB video. *TOG*, 2020. 2
- [86] Jikai Wang, Qifan Zhang, Yu-Wei Chao, Bowen Wen, Xiaohu Guo, and Yu Xiang. HO-Cap: A capture system and dataset for 3D reconstruction and pose tracking of hand-object interaction. *arXiv:2406.06843*, 2024. 2
- [87] Yangang Wang, Jianyuan Min, Jianjie Zhang, Yebin Liu, Feng Xu, Qionghai Dai, and Jinxiang Chai. Video-based hand manipulation capture through composite motion control. *TOG*, 2013.
- [88] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. BundleSDF: Neural 6-DoF tracking and 3D reconstruction of unknown objects. In *CVPR*, 2023. 2
- [89] Jane Wu, Georgios Pavlakos, Georgia Gkioxari, and Jitendra Malik. Reconstructing hand-held objects in 3D. *arXiv:2404.06507*, 2024. 2, 6, 8
- [90] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. SnowflakeNet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *ICCV*, 2021. 5
- [91] 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. 2

- [92] 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](#)
- [93] 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. [5](#)
- [94] 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 *CVPR*, 2022. [2](#)
- [95] Yufei Ye, Abhinav Gupta, and Shubham Tulsiani. What’s in your hands? 3D reconstruction of generic objects in hands. In *CVPR*, 2022. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [8](#)
- [96] Yufei Ye, Poorvi Hebbbar, Abhinav Gupta, and Shubham Tulsiani. Diffusion-guided reconstruction of everyday hand-object interaction clips. In *ICCV*, 2023. [1](#), [2](#), [3](#)
- [97] Frank Yu, Mathieu Salzmann, Pascal Fua, and Helge Rhodin. PCLs: Geometry-aware neural reconstruction of 3D pose with perspective crop layers. In *CVPR*, 2021. [6](#)
- [98] Shanxin Yuan, Guillermo Garcia-Hernando, Björn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, Junsong Yuan, Xinghao Chen, Guijin Wang, Fan Yang, Kai Akiyama, Yang Wu, Qingfu Wan, Meysam Madadi, Sergio Escalera, Shile Li, Dongheui Lee, Iason Oikonomidis, Antonis Argyros, and Tae-Kyun Kim. Depth-based 3D hand pose estimation: From current achievements to future goals. In *CVPR*, 2018. [2](#)
- [99] Chenyangguang Zhang, Yan Di, Ruida Zhang, Guangyao Zhai, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. DDF-HO: hand-held object reconstruction via conditional directed distance field. In *NeurIPS*, 2024. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [8](#)
- [100] Jinglei Zhang, Jiankang Deng, Chao Ma, and Rolandos Alexandros Potamias. HaWoR: World-space hand motion reconstruction from egocentric videos. In *CVPR*, 2025. [2](#)
- [101] Zhifan Zhu and Dima Damen. Get a grip: Reconstructing hand-object stable grasps in egocentric videos. *arXiv:2312.15719*, 2023. [1](#), [2](#)
- [102] Christian Zimmermann and Thomas Brox. Learning to estimate 3D hand pose from single RGB images. In *ICCV*, 2017. [2](#)

## Appendix

In appendix, we provide more details of our model architectures and additional results. We first present details of our model architecture in Section A. Then in Section B, we discuss additional experimental results.

### A. Network Architecture

#### A.1. PointNet encoder

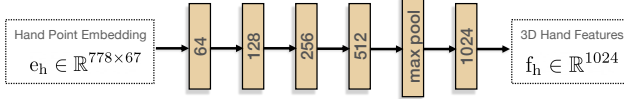


Figure 6. Network architecture used for our PointNet encoder. We use five fully-connected layers for efficiency. The number in the box denotes the dimension of features.

As shown in Figure 6, we employ five fully-connected layers to construct an efficient PointNet encoder. It takes the point clouds transformed into different hand coordinate systems ( $e_h \in \mathbb{R}^{778 \times 67}$ ) and outputs 1024-dimensional global features  $f_h \in \mathbb{R}^{1024}$  for the sparse point cloud decoder.

#### A.2. Dense point cloud decoder

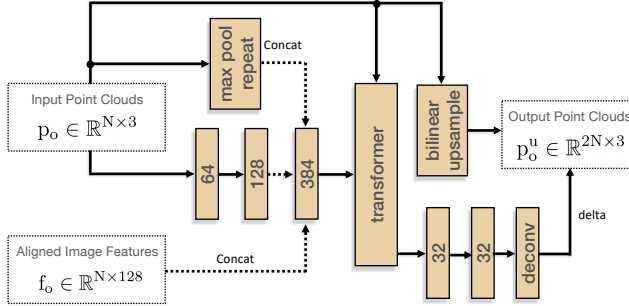


Figure 7. Network architecture used for reconstructing dense object point clouds. Given the input point clouds and its corresponding image features, the network performs point clouds upsampling. The number in the box denotes the dimension of features.

Figure 7 illustrates the network architecture for reconstructing dense object point clouds. Given input point clouds  $p_o \in \mathbb{R}^{N \times 3}$  and aligned image features  $f_o \in \mathbb{R}^{N \times 128}$ , we first apply convolution layers and max pooling to  $p_o$ , then concatenate the processed features with  $f_o$ . As discussed in the main paper, we perform the same operations for predicted hand vertices  $v_h$ , equipping each object point with its surrounding hand context. Next, we upsample the input object point clouds via bilinear interpolation and employ a transformer model to predict 3D offsets for the initially upsampled object points. Our model consists of two consecutive blocks, as shown in Figure 7, which progressively upsample sparse point clouds  $p_o^s \in \mathbb{R}^{2048 \times 3}$  to dense object point clouds  $p_o^d \in \mathbb{R}^{16384 \times 3}$ .

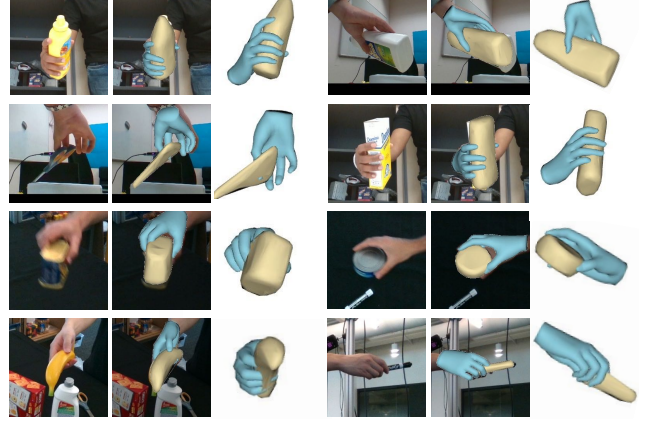


Figure 8. Qualitative results of reconstructed object meshes produced by our approach on HO3D [28] and DexYCB [9] datasets.

Table 7. Comparison with previous methods on DexYCB.

| Methods          | mesh         | FS@5 $\uparrow$ | FS@10 $\uparrow$ | CD $\downarrow$ | Cr $\uparrow$ | Pd $\downarrow$ |
|------------------|--------------|-----------------|------------------|-----------------|---------------|-----------------|
| R1 GF [38]       | $\checkmark$ | 0.39            | 0.66             | 4.5             | 0.96          | 0.92            |
| R2 AlignSDF [14] | $\checkmark$ | 0.41            | 0.68             | 3.9             | 0.97          | 1.08            |
| R3 gSDF [16]     | $\checkmark$ | 0.44            | 0.71             | 3.4             | 0.95          | 0.94            |
| R4 D-SCO [25]    | $\times$     | <b>0.63</b>     | 0.82             | 1.3             | -             | -               |
| R5 HORT (Ours)   | $\times$     | <b>0.63</b>     | <b>0.85</b>      | <b>1.1</b>      | <b>0.98</b>   | <b>0.90</b>     |
| R6 HORT (Ours)   | $\checkmark$ | 0.62            | <b>0.85</b>      | <b>1.1</b>      | <b>0.98</b>   | <b>0.90</b>     |

Table 8. Ablation studies of hand accuracy on DexYCB dataset.

| Methods                    | Noise | $E_h \downarrow$ | FS@5 $\uparrow$ | FS@10 $\uparrow$ | CD $\downarrow$ | $C_r \uparrow$ | $P_d \downarrow$ |
|----------------------------|-------|------------------|-----------------|------------------|-----------------|----------------|------------------|
| HORT (Ours) $\sigma = 0.0$ | 0.00  | <b>0.64</b>      | <b>0.88</b>     | <b>1.0</b>       | <b>0.98</b>     | <b>0.88</b>    |                  |
| HORT (Ours) $\sigma = 0.1$ | 7.67  | 0.60             | 0.85            | 1.3              | 0.96            | 0.93           |                  |
| HORT (Ours) $\sigma = 0.5$ | 36.11 | 0.54             | 0.79            | 1.6              | 0.91            | 1.02           |                  |

## B. Experimental Results

### B.1. Object mesh reconstruction results

When needed, we can easily convert our reconstructed object point clouds into meshes. As shown in Figure 8, we employ the alpha shapes algorithm [22], as implemented in the Open3D library, to generate high-fidelity object meshes. Table 7 presents our quantitative analysis of point cloud and mesh predictions. Here, we use predicted hand poses for different approaches on DexYCB [9] dataset. Compared to evaluations on point clouds in R5, R6 evaluates object meshes and achieves similar quantitative performance, further demonstrating the flexibility and efficiency of our point cloud representation. Moreover, compared to previous methods, HORT produces more physically plausible hand-object configurations and achieves better results in terms of Contact Ratio (Cr) and Penetration Depth (Pd).

### B.2. Comparison for 3D reconstruction densities

Figure 9 qualitatively compares our reconstructed sparse and dense object point clouds. We observe that our reconstructed sparse and dense point clouds are consistent in the general shape of the object. Our dense point clouds contain



Figure 9. Qualitative comparison of our reconstructed sparse and dense object point clouds on HO3D and DexYCB datasets.

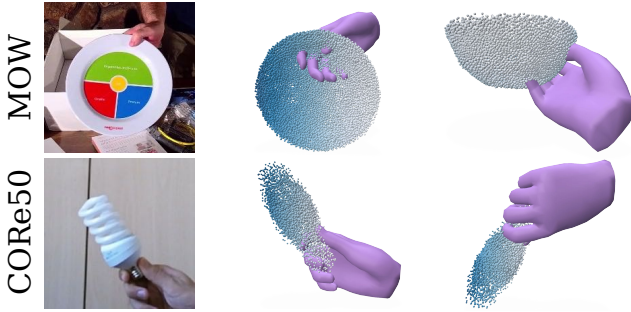


Figure 10. Failure cases on MOW [7] and CORE50 [50] images. more surface details for the manipulated object.

### B.3. Impact of imperfect hand poses

Table 8 ablates the impact of hand accuracy to our model by gradually adding Gaussian noise to ground-truth hand poses. As a result, the hand joint error  $E_h$  increases from 7.67mm to 36.11mm with more noises. Our HORT model is robust to noisy hand poses and can still reconstruct plausible 3D hand-held objects.

### B.4. Additional qualitative results

In this section, we present additional qualitative examples in Figure 11 to demonstrate that our approach produces high-quality 3D reconstructions across various challenging scenes. Our model performs well on both the synthetic ObMan [32] dataset and the real-world HO3D [28] and DexYCB [9] datasets. Furthermore, we show that our model can generate reliable predictions on in-the-wild CORE50 [50] images, highlighting the ability of our HORT model to effectively generalize to diverse object instances and textures in unconstrained environments. Figure 12 demonstrates that our reconstructed results are temporally consistent and robust to hand occlusion.

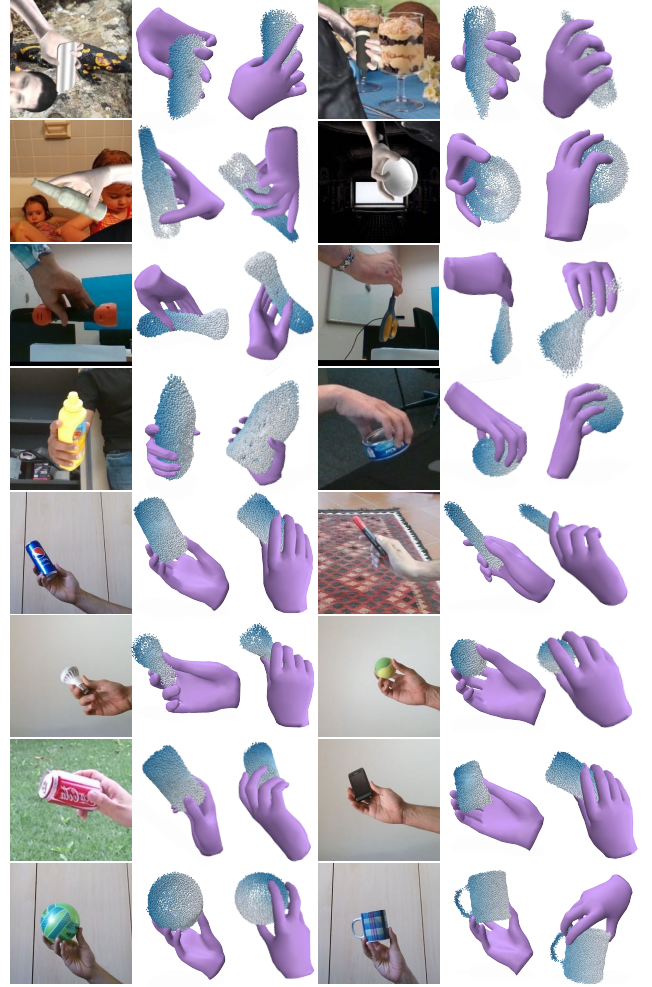


Figure 11. Qualitative results of HORT on synthetic ObMan [32], indoor HO3D [28] and DexYCB [9], and in-the-wild CORE50 [50] images. Our model shows impressive results on all these domains.

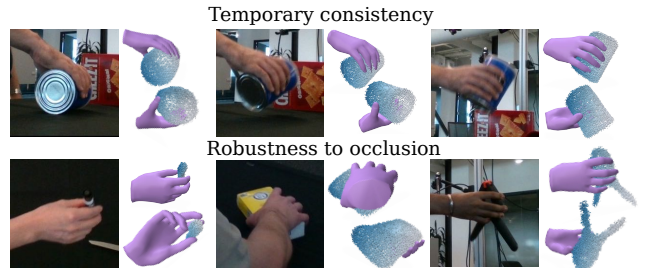


Figure 12. Qualitative analysis on DexYCB datasets.

### B.5. Failure cases analysis

While HORT achieves impressive results, it struggles to accurately infer hand-object configurations when the hand is heavily occluded (Figure 10, top) and to recover fine-grained geometry for rare objects (Figure 10, bottom). Scaling training data with a broader range of objects and grasping poses could help mitigate these limitations.