Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

Robo3R: Enhancing Robotic Manipulation with Accurate Feed-Forward 3D Reconstruction

Sizhe Yang[1,2] Linning Xu[1,2,†] Hao Li[1,3] Juncheng Mu[1,4] Jia Zeng[1] Dahua Lin[1,2] Jiangmiao Pang[1,†]
[1]Shanghai AI Laboratory [2]The Chinese University of Hong Kong
[3]University of Science and Technology of China [4]Tsinghua University
[†] Corresponding authors
Project page: https://yangsizhe.github.io/robo3r/

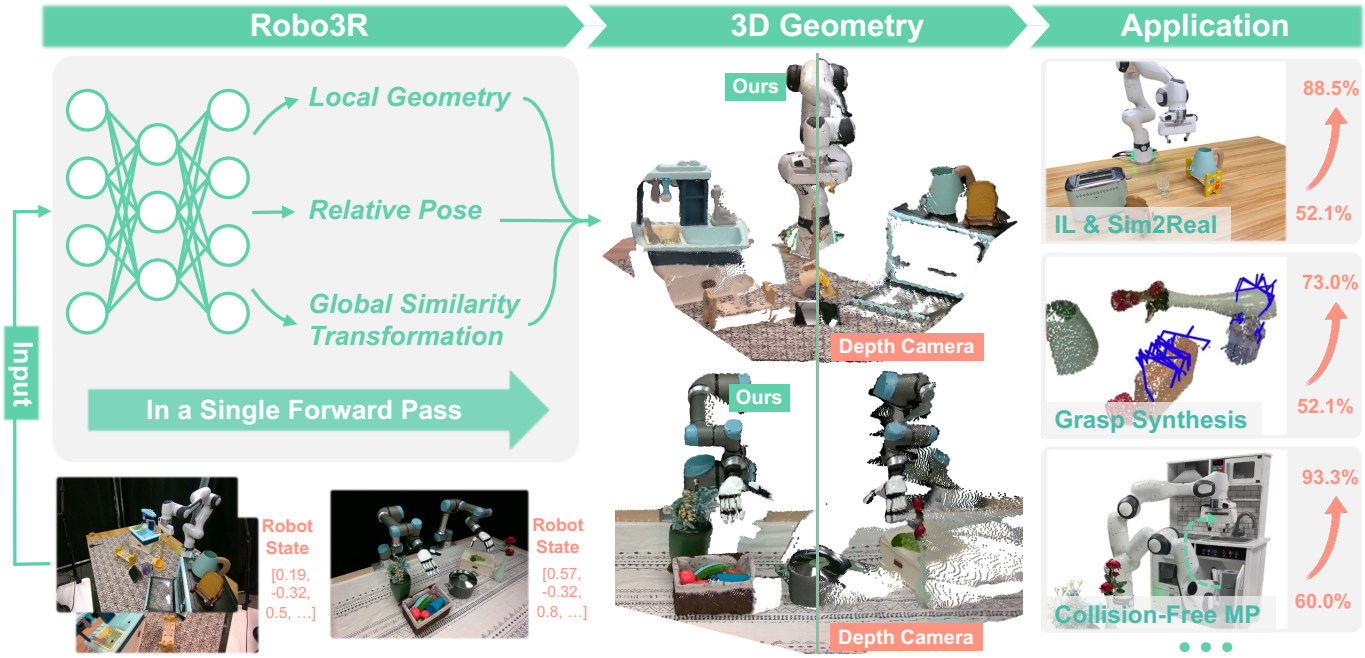

Fig. 1: **Overview. Robo3R** enables manipulation-ready 3D reconstruction from RGB frames in real time. By achieving accurate metric-scale 3D geometry in the canonical robot frame, Robo3R **eliminates the need for depth sensors and calibration, while improving accuracy and robustness** in challenging manipulation scenarios. These features lead to notable improvements in downstream applications such as imitation learning, sim-to-real transfer, grasp synthesis, and collision-free motion planning.

***Abstract*—3D spatial perception is fundamental to generalizable robotic manipulation, yet obtaining reliable, high-quality 3D geometry remains challenging. Depth sensors suffer from noise and material sensitivity, while existing reconstruction models lack the precision and metric consistency required for physical interaction. We introduce Robo3R, a feed-forward, manipulation-ready 3D reconstruction model that predicts accurate, metric-scale scene geometry directly from RGB images and robot states in real time. Robo3R jointly infers scale-invariant local geometry and relative camera poses, which are unified into the scene representation in the canonical robot frame via a learned global similarity transformation. To meet the precision demands of manipulation, Robo3R employs a masked point head for sharp, fine-grained point clouds, and a keypoint-based Perspective-n-Point (PnP) formulation to refine camera extrinsics and global alignment. Trained on Robo3R-4M, a curated large-scale synthetic dataset with four million high-fidelity annotated frames, Robo3R consistently outperforms state-of-the-art reconstruction methods and depth sensors. Across downstream tasks including imitation learning, sim-to-real transfer, grasp synthesis, and collision-free motion planning, we observe consistent gains in performance, suggesting the promise of this alternative 3D sensing module for robotic manipulation.**

I. Introduction

In unstructured environments, generalizable robotic manipulation relies heavily on robust spatial perception of the physical world. The perception of the external environment for robotic systems is generally categorized by input modality into 2D-based and 3D-based types. While RGB images are readily accessible, 2D-based policies often require extensive training data and frequently struggle with complex tasks [51]. In contrast, 3D-based policies are more data-efficient and

generalizable. Furthermore, for geometry-centric capabilities, such as grasp synthesis [7, 54] and collision-free motion planning [3, 45, 29], explicit 3D input is typically indispensable for ensuring accurate physical grounding.

However, acquiring 3D modalities is significantly more challenging than acquiring RGB images. Currently, 3D data for robotic manipulation is primarily obtained using depth cameras, which often produce depth maps of limited quality. Common depth cameras, whether stereo-based (e.g., RealSense D455) or time-of-flight based (e.g., Azure Kinect), are susceptible to noise and inaccuracies. Moreover, the quality of depth maps degrades drastically when encountering transparent or reflective objects, or under adverse lighting conditions.

Recently, the fields of depth estimation [46, 34, 35, 41, 2] and feed-forward 3D reconstruction [36, 31, 13, 39] have witnessed rapid advancement. However, despite their promise for enhancing spatial perception in robotics, most existing feed-forward reconstruction models still lack manipulation-level geometric precision and reliable metric scale, limiting their applicability in real-world robotic manipulation.

To address these challenges, we propose **Robo3R**, a feed-forward 3D reconstruction model designed specifically for robotic manipulation, as shown in Fig. 1. Robo3R fuses RGB images with robot states and processes the fused features using the Alternating-Attention mechanism, which facilitates efficient information propagation within and across frames. Robo3R incorporates several specialized decoding heads. The proposed masked point head decomposes dense point prediction into depth, normalized image coordinate, and mask prediction, mitigating over-smoothing and producing sharp, fine-grained geometric details via unprojection and masking. The relative pose head predicts relative poses to register points across multiple views, while the similarity transformation (S.T.) tokens extract global similarity transformations, mapping points into metric-scale 3D geometry within the canonical robot frame. An extrinsic estimation module extracts robot keypoints and estimates camera extrinsics by solving the Perspective-n-Point (PnP) problem [15], further refining the similarity transformation. Compared to previous feed-forward reconstruction models, Robo3R explicitly incorporates robot priors, significantly enhancing reconstruction fidelity. We train Robo3R on our curated large-scale high-fidelity synthetic dataset that contains four million frames. Thanks to the dataset's high diversity and photorealism, Robo3R successfully transfers to real-world manipulation scenarios.

We validate Robo3R through extensive experiments, demonstrating that it outperforms state-of-the-art feed-forward reconstruction models and depth sensors in terms of 3D geometry quality. The accurate 3D geometry produced by Robo3R enables improved performance in downstream applications, including real-world imitation learning, sim-to-real transfer, grasp synthesis, and collision-free motion planning. Crucially, our results highlight Robo3R's superior robustness in challenging scenarios: it effectively handles transparent, reflective, and tiny objects that typically hinder depth cameras.

In summary, the contributions of this paper are threefold:

1) We introduce **Robo3R-4M**, a large-scale synthetic dataset and corresponding data pipeline designed for perception and reconstruction in robotic manipulation scenarios. The dataset comprises four million frames and features diverse assets, extensive randomization, rich modalities, and annotations.
2) We propose **Robo3R**, a feed-forward 3D reconstruction model tailored for robotic manipulation, which achieves high-fidelity depth estimation, precise camera parameter prediction, accurate metric scaling, and maintains a canonical coordinate system in real time.
3) We conduct extensive qualitative and quantitative experiments validating Robo3R as a superior alternative to depth cameras. The results demonstrate that Robo3R produces higher-quality 3D representation, exhibits greater robustness to varying object materials and challenging scenarios, and significantly enhances spatial perception for robotic manipulation.

II. Related Work

Feed-forward 3D reconstruction. In recent years, 3D reconstruction has rapidly advanced, particularly with the adoption of feed-forward neural networks. DUSt3R [36] has first shown promising results in this direction, predicting point maps from two images. VGGT [31] takes a further step by enabling 3D reconstruction from one, a few, or even hundreds of input views. Several studies address the reconstruction of dynamic scenes from videos [55, 32, 52, 21, 28, 8]. π^3 [37] utilizes a permutation-equivariant architecture to estimate affine-invariant camera poses and scale-invariant local point maps. MapAnything [13] supports optional geometric inputs and achieves metric scale reconstruction. DepthAnything3 [18] further simplifies feed-forward 3D reconstruction by using a single plain transformer and a minimal set of depth-ray prediction targets. Nevertheless, these approaches mainly focus on scene-level reconstruction and exhibit limited accuracy in fine-grained geometry and scale, which are crucial for robotic manipulation. Our work addresses these limitations through the synthesis of high-quality data and careful model design.

3D representation for manipulation. Compared to RGB images, 3D representation provides better physical grounding and are therefore widely used in robotic manipulation. Several studies have employed 3D representation as model inputs to enable efficient and generalizable policy learning [51, 44, 9, 27, 50, 40, 12, 10, 11, 43, 22, 24]. It has also been shown that 3D representation is more suitable than RGB images for sim-to-real transfer of manipulation policies [23, 48, 26, 20, 49]. Moreover, grasp synthesis models typically rely on 3D representation to model the scene [7, 54, 38, 33, 17, 42, 47], thereby enabling the prediction of grasp poses grounded in 3D geometry. Collision-free motion planning in real-world environments also requires 3D representation [29, 3, 45, 14], as explicit 3D data provide superior spatial and geometric information for obstacle avoidance. Recently, several studies leverage implicit 3D representation to enhance the performance of manipulation policies [16, 25, 53, 19]. We evaluate

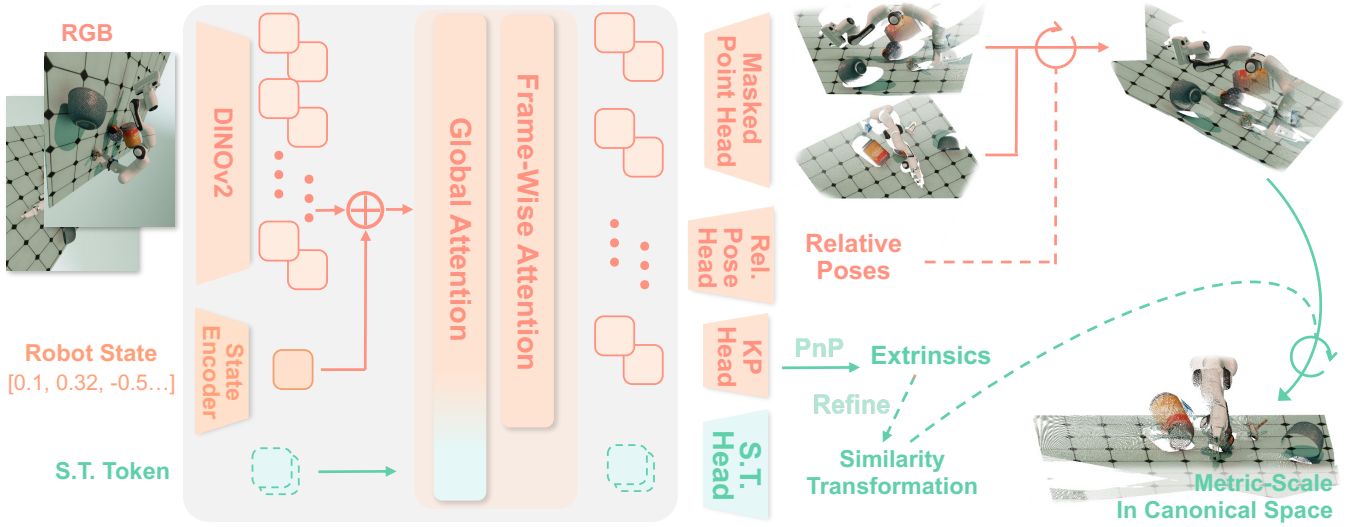

Fig. 2: **Method Overview**. RGB images and robot states are encoded and fused. The transformer backbone processes the resulting features through alternating global and frame-wise attention. The masked point head decodes scale-invariant local geometry, while the relative pose head outputs relative poses for registering points across multiple views. S.T. tokens read out the global similarity transformation, which maps the points into metric-scale 3D geometry in the canonical robot frame.

the quality and usability of the 3D representation reconstructed by Robo3R in downstream applications such as imitation learning, sim-to-real transfer, grasp synthesis, and collision-free motion planning.

III. Methodology

An overview of Robo3R is shown in Fig. 2. In this section, we describe Robo3R in detail. We begin with a brief formulation of the task and 3D representation (Section III-A). Next, we elaborate on the carefully designed model architecture (Section III-B). We then discuss the curation of Robo3R-4M, a large-scale, high-quality synthetic dataset that supports the training of Robo3R (Section III-C). Finally, we provide details about the training objectives (Section III-D).

A. Formulation

1) Task Definition: The task we address is metric-scale and fine-grained 3D reconstruction from sparse views in robotic manipulation scenarios. Specifically, the input consists of monocular or binocular RGB images $\{I_i\}_{i=1}^{N}, I_i \in \mathbb{R}^{H\times W\times 3}, N \in \{1,2\}$, and the robot state $\mathbf{J} \in \mathbb{R}^{Q}$, where $\mathbf{J}$ denotes the robot's joint angles and Q is the number of joints. A feed-forward neural network performs the 3D reconstruction and outputs a comprehensive set of 3D attributes, including depth maps $D \in \mathbb{R}^{H\times W}$, normalized image coordinates $C \in \mathbb{R}^{H\times W\times 2}$ (which represent the 2D pixel positions projected onto the normalized image plane at unit depth), relative camera translation $\mathbf{t}_{\text{rel}} \in \mathbb{R}^3$, relative camera rotation $\mathbf{R}_{\text{rel}} \in \mathbb{R}^{3\times 3}$, and a global similarity transformation $\mathbf{S} \in \mathbb{R}^{4\times 4}$.

2) Scale-Invariant Local 3D Representation: Directly predicting metric-scale 3D points in the world coordinate system is challenging. Therefore, we first predict a scale-invariant local 3D representation in the camera coordinate system. Given normalized image coordinates $\mathbf{c} = (x, y) \in \mathbb{R}^2$ and depth $d \in \mathbb{R}$, we derive the local point cloud $\mathbf{P}_{\text{local}} \in \mathbb{R}^3$ via unprojection. Since the normalized coordinates represent the 2D projection on the unit depth plane ($d = 1$), the 3D position can be recovered by scaling the direction vector $(x, y, 1)$ by the depth d:

$$\mathbf{P}_{\text{local}} = [x \cdot d,\ y \cdot d,\ d]^\top. \tag{1}$$

This formulation decouples ray direction from depth estimation, enabling the network to learn geometric structure in a local canonical space before transforming it to the global frame. To address scale ambiguity, Robo3R predicts scale-invariant local points. Specifically, before computing the loss between the predicted local points $\hat{\mathbf{P}}_{\text{local}}$ and the ground truth $\mathbf{P}^{\text{gt}}_{\text{local}}$, we multiply $\hat{\mathbf{P}}_{\text{local}}$ by a scale factor s that is determined by aligning the prediction with the ground truth.

3) Metric-Scale Geometry in the Canonical Frame: After obtaining the scale-invariant local points $\mathbf{P}_{\text{local}}$, we register the local points from multiple views $\{\mathbf{P}^i_{\text{local}}\}_{i=1}^{N}$ via the predicted relative camera translation $\mathbf{t}_{\text{rel}}$ and rotation $\mathbf{R}_{\text{rel}}$:

$$\mathbf{P}_{\text{reg}} = \left\{ \mathbf{R}^i_{\text{rel}}\mathbf{P}^i_{\text{local}} + \mathbf{t}^i_{\text{rel}} \mid i = 1, \ldots, N \right\}, \tag{2}$$

where i indexes the N different views. Then, we transform the registered points $\mathbf{P}_{\text{reg}}$ into the metric-scale geometry in the canonical robot frame $\mathbf{P}_{\text{cano}}$ via the predicted global similarity transformation S:

$$\mathbf{P}_{\text{cano}} = \left\{ [\mathbf{p};\, 1]\, \mathbf{S}^\top \mid \mathbf{p} \in \mathbf{P}_{\text{reg}} \right\}. \tag{3}$$

B. Model Architecture

1) Encoders: The inputs to Robo3R are images and robot states. We employ DINOv2 ViT-L to encode images from N views $\{I_i\}_{i=1}^{N}, I_i \in \mathbb{R}^{H\times W\times 3}$ into patch features $\{F_{I,i}\}_{i=1}^{N}, F_{I,i} \in \mathbb{R}^{\frac{H}{14}\times\frac{W}{14}\times 1024}$. The robot states are projected to state features $F_J \in \mathbb{R}^{1024}$ using a multilayer perceptron

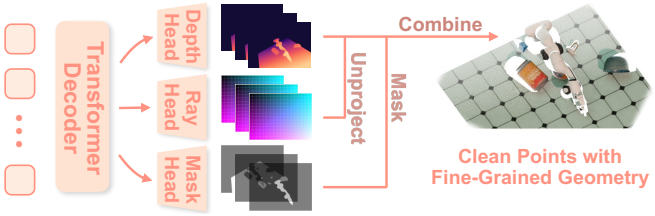

Fig. 3: **Masked point head.** To address the over-smoothing problem for dense prediction, we propose a masked point head that decomposes point prediction into depth, normalized image coordinate, and mask predictions. Through unprojection, masking, and combination, we obtain sharp points with fine-grained geometric details.

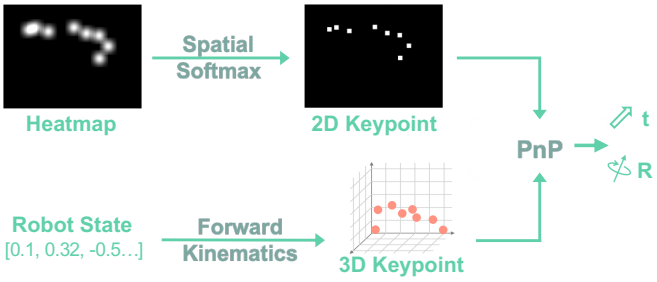

Fig. 4: **Extrinsic estimation module.** The extrinsic estimation module extracts robot keypoints and accurately estimates the camera extrinsics by solving the Perspective-n-Point (PnP) problem; the camera extrinsics are used to refine the global similarity transformation.

(MLP) with GeLU activations. The image and state features are then fused via element-wise addition to obtain the combined features $F \in \mathbb{R}^{N \times \frac{H}{14} \times \frac{W}{14} \times 1024}$. Then, we append learnable similarity transformation (S.T.) tokens to F, which serve as the input to the transformer backbone.

2) Transformer Backbone: We employ a transformer backbone based on the Alternating-Attention mechanism [31]. Specifically, we stack 18 alternating global and frame-wise attention blocks to enable efficient propagation of information both within and across frames.

3) Prediction Heads:

Masked point head. Over-smoothing is a persistent issue in dense prediction, which could result in blurry edges and loss of detail in point clouds. To address this problem, we decouple dense point prediction into depth, image coordinate, and mask components, as illustrated in Fig. 3. The masked point head consists of three branches dedicated to the robot, objects, and background. Each branch contains a depth head, a ray head, and a mask head, and produces a point map by unprojecting the depth along the corresponding ray directions. The robot, object, and background masks then extract region-specific points from their respective point maps, which are aggregated to reconstruct the full scene point cloud. Specifically, the masked point head consists of a five-layer transformer decoder, a depth head, a ray head, and a mask head. Patch-wise features produced by the transformer backbone are processed by the transformer decoder. The outputs are then decoded into depth, image coordinates, and masks using the MLP heads followed by pixel shuffle operations.

Relative pose head. Adapted from Dong et al. [5] and Wang et al. [37], the head processes outputs of the transformer backbone with a transformer decoder, two residual convolution blocks and adaptive average pooling, followed by an MLP. It predicts translation as a 3D vector and rotation in a 9D representation. The 9D rotation is reshaped into a 3×3 matrix and orthogonalized via SVD, ensuring a valid rotation.

Similarity transformation head. The latent representation of the similarity transformation extracted by the S.T. tokens is decoded by a network with the same architecture as the relative pose head, together with an MLP that predicts the scale. This process yields the rigid transformation $\mathbf{T}$ and the scale s, from which the similarity transformation $\mathbf{S}$ can be obtained:

$$\mathbf{S} = s \cdot \mathbf{T}. \tag{4}$$

Extrinsic estimation module and keypoint head. Although Robo3R directly predicts the camera extrinsics, which are part of the similarity transformation, we additionally develop a more accurate and robust extrinsic estimation module, as illustrated in Fig. 4. The results obtained from this module can further refine the similarity transformation. We compute 3D keypoints as the frame origin of each link by applying forward kinematics to the robot's joint positions. During data synthesis, these keypoints are projected onto the image plane via camera intrinsics and extrinsics to obtain 2D keypoints, each rendered as a Gaussian blob to produce a per-keypoint heatmap. During inference, Soft-Argmax is applied to heatmaps to extract 2D keypoints as a weighted average of pixel coordinates, yielding floating-point values that enable sub-pixel accuracy. The 3D keypoints and per-keypoint heatmaps are index-aligned, enabling matching between 3D and 2D keypoints. Robo3R predicts the pixel coordinates of these keypoints with a keypoint head, and the camera extrinsics are subsequently estimated by solving a PnP problem. In the keypoint head, patch features are first processed by a transformer decoder, then decoded by an MLP followed by a pixel shuffle operation to generate a keypoint heatmap $M_{kp} \in \mathbb{R}^{H \times W \times N_{kp}}$, where N_{kp} is the number of keypoints. Finally, we employ a differentiable Soft-Argmax operator to extract 2D keypoint coordinates $C_{kp} \in \mathbb{R}^{N_{kp} \times 2}$ from the heatmaps with sub-pixel accuracy.

C. Synthetic Data Pipeline

We curate a large-scale synthetic dataset, Robo3R-4M, using NVIDIA Isaac Sim as both the physics and rendering engine. We utilize a diverse set of assets, including 16,911 objects sourced from DTC [6] and Objaverse [4], 4,710 textures, and 6,512 environment maps. Extensive domain randomization is applied to various factors, including robot behaviors, camera extrinsics and intrinsics, object instances and poses, table instances and poses, background, and lighting conditions. Multiple data modalities are recorded, including RGB images, depth images, semantic masks, robot states,

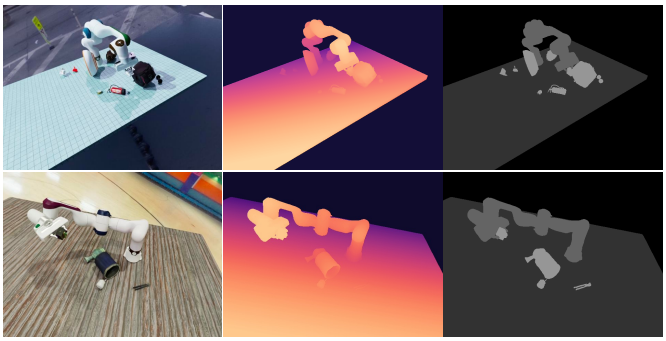

Fig. 5: **Data samples.** The dataset showcases a diverse array of assets with extensive randomization, encompassing rich modalities and comprehensive annotations.

camera intrinsics and extrinsics. Representative data samples are illustrated in Fig. 5. Robo3R-4M consists of 100,000 scenes and four million photorealistic, high-quality frames, which support the training of Robo3R. See more details about synthetic data generation in Appendix A.

D. Training Objectives

Point loss. Instead of directly supervising the depth and normalized image coordinates, we supervise the point cloud obtained via unprojection. The point loss is defined as follows:

$$\mathcal{L}_{\text{point}} = \frac{1}{3HW} \sum_{i=1}^{H \times W} \left\| s \cdot \hat{\mathbf{p}}_i - \mathbf{p}_i \right\|_1, \tag{5}$$

where s is a scale factor determined by aligning the predicted point maps with the ground truth.

Normal loss. To further supervise the geometric consistency of the predicted point clouds, we supervise surface normals during training. For each point in the predicted point map, the normal vector is computed by taking the cross product of vectors connecting the point to its neighboring points. The normal loss is defined as:

$$\mathcal{L}_{\text{normal}} = \frac{1}{K} \sum_{k=1}^{K} \Delta\theta_k, \tag{6}$$

where K denotes the number of valid normal pairs, and $\Delta\theta_k$ is the angle between the predicted and ground-truth normals at location k. The angle is computed as

$$\Delta\theta_k = \arctan 2 \left(\left\| \hat{\mathbf{n}}_k \times \mathbf{n}_k \right\|, \hat{\mathbf{n}}_k \cdot \mathbf{n}_k \right), \tag{7}$$

where $\hat{\mathbf{n}}_k$ and $\mathbf{n}_k$ denote the normal vectors estimated from the predicted and ground-truth points, respectively.

Mask loss. In the masked point head, we employ pixel-wise masks to obtain clean points with fine-grained geometry. Specifically, we predict separate masks for the robot, objects, and background. The mask loss is defined as the binary cross-entropy between the predicted masks $\hat{m}_i$ and the ground-truth masks m_i:

$$\mathcal{L}_{\text{mask}} = \frac{1}{HW} \sum_{i=1}^{H \times W} \text{BCE} \left(\hat{m}_i, m_i \right). \tag{8}$$

Relative pose loss. The relative pose head predicts a camera translation $\hat{\mathbf{t}}$ and rotation $\hat{\mathbf{R}}$ for each view. The relative pose from view j to view i is computed as:

$$\hat{\mathbf{R}}_{\text{rel}} = \hat{\mathbf{R}}_i^{-1} \hat{\mathbf{R}}_j, \tag{9}$$

$$\hat{\mathbf{t}}_{\text{rel}} = \hat{\mathbf{R}}_i^{-1} (\hat{\mathbf{t}}_j - \hat{\mathbf{t}}_i), \tag{10}$$

where $\hat{\mathbf{R}}_i$ and $\hat{\mathbf{t}}_i$ denote the predicted rotation and translation of view i. The relative pose loss is defined as:

$$\mathcal{L}_{\text{rel}} = \alpha \cdot \text{Huber} \left(\hat{\mathbf{t}}_{\text{rel}}, \mathbf{t}_{\text{rel}} \right) + \text{Angle} \left(\hat{\mathbf{R}}_{\text{rel}}, \mathbf{R}_{\text{rel}} \right), \tag{11}$$

where $\text{Huber}(\cdot, \cdot)$ denotes the Huber loss for translation, $\text{Angle}(\cdot, \cdot)$ computes the rotation angular error, and α is a balancing weight.

Similarity transformation loss. The similarity transformation head predicts the global similarity transformation, which consists of a scale factor $\hat{s}$, as well as the camera translation $\hat{\mathbf{t}}_{\text{abs}}$ and rotation $\hat{\mathbf{R}}_{\text{abs}}$ for the first view. The similarity transformation loss supervises these components and is defined as:

$$\begin{aligned} \mathcal{L}_{\text{ST}} = \beta_1 \cdot \text{Huber} \left(\hat{s}, s \right) + \beta_2 \cdot \text{Huber} \left(\hat{\mathbf{t}}_{\text{abs}}, \mathbf{t}_{\text{abs}} \right) \\ + \text{Angle} \left(\hat{\mathbf{R}}_{\text{abs}}, \mathbf{R}_{\text{abs}} \right). \end{aligned} \tag{12}$$

Keypoint loss. The keypoint loss jointly supervises the keypoint heatmap and the 2D keypoint coordinates, which are extracted from the heatmap using a differentiable Soft-Argmax operator. The loss is defined as:

$$\mathcal{L}_{\text{kp}} = \gamma \cdot \left\| \hat{M}_{\text{kp}} - M_{\text{kp}} \right\|_1 + \left\| \hat{C}_{\text{kp}} - C_{\text{kp}} \right\|_1, \tag{13}$$

where $\hat{M}_{\text{kp}}$ and M_{kp} denote the predicted and ground-truth keypoint heatmaps, $\hat{C}_{\text{kp}}$ and C_{kp} are the predicted and ground-truth keypoint coordinates, and γ is balancing weights.

Overall. Robo3R is trained end-to-end by minimizing the multi-task loss $\mathcal{L}$, which is the weighted sum of the point loss, normal loss, mask loss, relative pose loss, similarity transformation loss, and keypoint loss:

$$\begin{aligned} \mathcal{L} = \lambda_1 \mathcal{L}_{\text{point}} + \lambda_2 \mathcal{L}_{\text{normal}} + \lambda_3 \mathcal{L}_{\text{mask}} \\ + \lambda_4 \mathcal{L}_{\text{rel}} + \lambda_5 \mathcal{L}_{\text{ST}} + \lambda_6 \mathcal{L}_{\text{kp}}, \end{aligned} \tag{14}$$

where $\lambda_1, \ldots, \lambda_6$ are the weights for each loss term. Additional training details are provided in Appendix B.

IV. Experiments

We conduct extensive experiments to evaluate the effectiveness of Robo3R, with respect to both reconstruction quality and performance on downstream tasks. Specifically, we seek to address the following questions:

1) How does Robo3R perform in terms of 3D reconstruction quality? In particular, does it achieve high-fidelity geometry, and does it accurately predict scale and camera pose?
2) How robust is Robo3R in challenging scenarios where depth cameras typically fail?
3) Can Robo3R be successfully applied to various downstream robotic manipulation applications?
4) Do the key design components effectively improve the performance of reconstruction?

A. 3D Reconstruction Quality

1) Quantitative Performance:

Benchmark. We constructed a benchmark for 3D reconstruction in robotic manipulation scenarios by rendering a photorealistic and diverse dataset, using objects, textures, and environment maps that are different from those used in training. The test set contains 2,000 scenes and 80,000 frames. The evaluation consists of two aspects: point map estimation and relative camera pose estimation. The metrics for point map estimation include scale-invariant point error and scale error. For relative camera pose estimation, the metrics include relative translation error (RTE), relative rotation error (RRE), relative translation accuracy (RTA) at a given threshold (e.g., RTA@0.03 for 0.03 meters), and relative rotation accuracy (RRA). As robots typically perceive the environment from sparse views, we evaluate point map estimation under monocular and binocular settings, and assess relative camera pose estimation under the binocular setting.

Baselines. We select four leading feed-forward 3D reconstruction models as baselines: VGGT [31], π^3 [37], DepthAnything3 (DA3) [18], and MapAnything (MA) [13]. Among them, MA and DA3 perform metric-scale reconstruction. Additionally, we consider MapAnything fine-tuned on the Robo3R-4M dataset (MA-FT) as a baseline.

Results. We evaluate Robo3R against strong baselines on the benchmark. Tab. I presents the quantitative results for point map estimation. Our method significantly outperforms all baselines across all metrics in both monocular and binocular settings. In the monocular case, Robo3R achieves a point error of 0.006, representing an order of magnitude improvement over the second-best method, π^3. Unlike the other models, which suffer from severe scale ambiguity (scale errors > 0.46), Robo3R effectively recovers metric geometry. This superiority persists in the binocular setting, where Robo3R further reduces errors, demonstrating robustness in 3D reconstruction for challenging manipulation scenarios. Additionally, Robo3R consistently outperforms the fine-tuned MapAnything model (MA-FT). This highlighting the benefits of our proposed model architecture. As shown in Tab. II, our method demonstrates exceptional precision in relative pose estimation. Robo3R achieves an RTE of 0.014 and an RRE of 0.013, which are approximately $8\times$ and $5\times$ lower than those of the best baseline, π^3, respectively. Furthermore, the high accuracy rates (RTA@0.03 of 0.951) confirm that Robo3R consistently provides reliable relative camera pose.

2) Qualitative Comparisons in Real-World Scenarios: We compare our method with the state-of-the-art feed-forward 3D reconstruction model π^3 [37], the depth completion model LingBot-Depth [30], as well as the depth camera. Compared to the other reconstruction models and the depth camera, Robo3R achieves significantly cleaner, more accurate 3D geometry with finer-grained details, and robustly handles challenging scenarios where the other methods fail, as illustrated in Fig. 6. Specifically, Robo3R is capable of reconstructing objects as narrow as 1.5 mm (spanning only 1 to 2 pixels in the image), whereas other methods, including depth cameras, fail to capture such fine geometry (row 1). Furthermore, Robo3R successfully handles reflective and transparent objects that blind depth sensors (row 2). Even in cluttered scenes that include bimanual robots with dexterous hands, Robo3R consistently produces accurate and clean point clouds (row 3).

TABLE I: **Results on point map estimation.** We report the scale-invariant point error and scale error for each method.

	Method	Point Err. ↓	Scale Err. ↓
Monocular	VGGT	0.126	0.663
	π^3	0.061	0.497
	DA3	0.075	0.506
	MA	0.078	0.467
	MA-FT	0.010	0.010
	Ours	**0.006**	**0.007**
Binocular	VGGT	0.220	0.619
	π^3	0.032	0.483
	DA3	0.042	0.719
	MA	0.076	0.540
	MA-FT	0.009	0.007
	Ours	**0.005**	**0.004**

TABLE II: **Results on relative camera pose prediction.** We report the relative translation error (RTE), relative rotation error (RRE), relative translation accuracy (RTA), and relative rotation accuracy (RRA) for each method.

Method	RTE ↓	RRE ↓	RTA@0.03 ↑	RRA@0.03 ↑
VGGT	0.373	0.316	0.014	0.052
π^3	0.116	0.073	0.110	0.245
MA	0.168	0.116	0.031	0.069
DA3	0.160	0.111	0.129	0.226
Ours	**0.014**	**0.013**	**0.951**	**0.899**

B. Downstream Robotic Manipulation

Overview of the manipulation benchmark. We evaluate the effectiveness of Robo3R for robotic manipulation on the real-world benchmark. The robotic platforms include a single-arm Franka Research 3 with a parallel gripper and a bimanual UR5e, each equipped with an XHand. RealSense D455 cameras are used to acquire RGB and depth images of the scene. Robo3R and the manipulation policy are deployed on an NVIDIA RTX 4090. The detailed inference speed of Robo3R is provided in Appendix C. The control frequency of the robotic system is set to 10 Hz. We conduct experiments across four applications: imitation learning, sim-to-real transfer, grasp synthesis, and collision-free motion planning. Additional details regarding the real-world manipulation experiments are provided in Appendix D.

1) Imitation Learning:

Experimental setup and baselines. Robo3R is evaluated on four tasks:

1) **Sweep Bean**: The gripper grasps the handle of a broom and sweeps beans on the table into a dustpan. The small size of the beans presents challenges for depth sensors and scene reconstruction.
2) **Insert Screw**: The robot inserts a screw into a hole where the hole's radius exceeds the screw's radius by

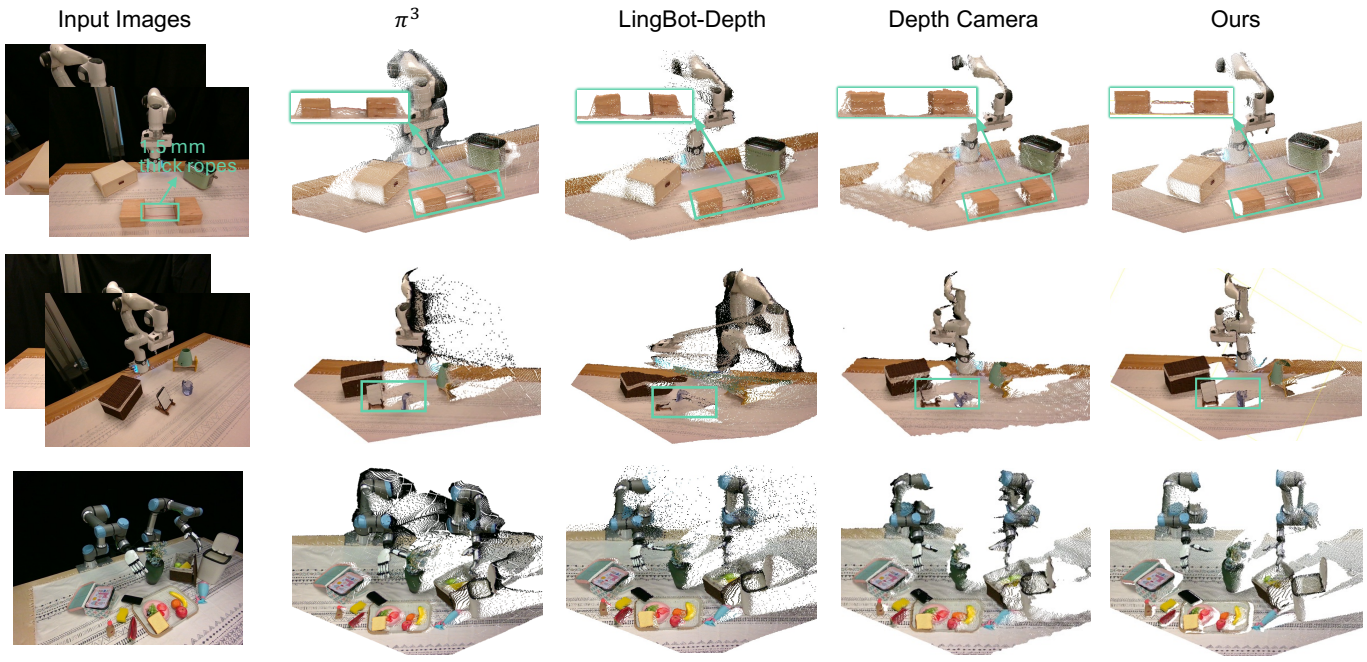

Fig. 6: **Qualitative Comparisons of 3D geometry.** We use a RealSense D455 as the depth camera and manually align the scale for point clouds reconstructed by π^3. We evaluate the methods on several challenging scenarios, including a tiny object (row 1), a scene with a mirror and a transparent cup (row 2), and a cluttered environment (row 3).

only 2 mm, requiring millimeter-level reconstruction and manipulation precision.

3) **Breakfast**: This is a long-horizon task in which the robot first picks up a slice of bread, places it in a toaster, presses the toaster button to toast the bread, then picks up a milk jug, pours milk into a glass, returns the jug to its original position, and finally pushes the glass forward.
4) **BiDex Pour**: This is a bimanual dexterous manipulation task, where the left hand holds a cup and the right hand holds a kettle to pour water into the cup.

We use Maniflow (MF) [44] as both a 2D- and 3D-based manipulation policy. Robo3R takes RGB images as input and produces point clouds, which are then used as input for the 3D-based MF. The baseline is the 2D-based MF, which takes RGB images as input. Additionally, as baselines, we use point clouds reconstructed by the other feed-forward 3D reconstruction models (Other FFs), as well as those obtained from a depth camera, as inputs for the 3D-based MF. π_0 [1], a state-of-the-art vision-language-action model, is also selected as a baseline.

Results. As can be seen in Tab. III, Robo3R combined with the 3D-based MF demonstrates superior or competitive performance compared to all baselines. Across all tasks, Robo3R (MF + Ours) consistently outperforms the 2D-based MF baseline and π_0, confirming the benefit of lifting 2D observations into 3D for manipulation policies. Furthermore, Robo3R is better than depth cameras in scenarios involving small objects or high precision. The "Other FFs" baseline failed to produce feasible actions (indicated by "-"), due to significant errors in scale and accuracy of the geometry, as well as the inability to effectively crop out cluttered background.

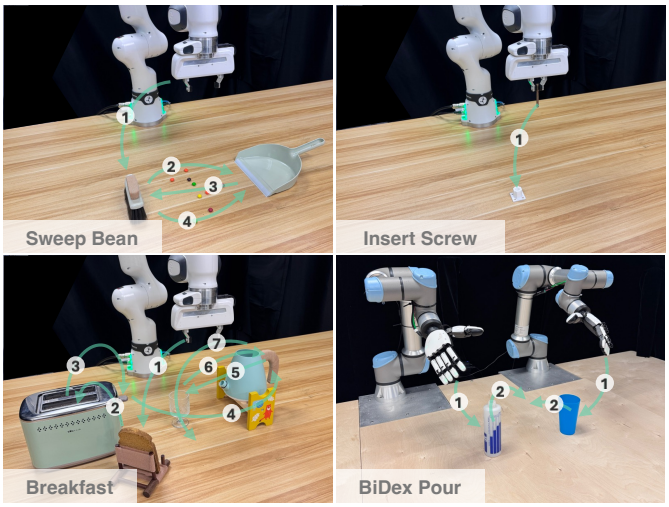

Fig. 7: **Imitation learning tasks.** We design four manipulation tasks for imitation learning evaluation: Sweep Bean, Insert Screw, Breakfast, and BiDex Pour.

2) Sim-to-Real Transfer:

Experimental setup and baselines. We collect 200 demonstrations for each of the "Push Cube" and "Pick Cube" tasks in NVIDIA Isaac Sim. The baseline methods acquire data using either RGB or depth cameras in simulation and deploy policies using the same modality in the real world. Robo3R reconstructs 3D geometry from RGB images in both simulation and the real world, using the resulting point clouds as inputs for policy training and deployment.

Results. The visualization of the visual gap between simu-

TABLE III: **Results on imitation learning.** We report the number of successes over the total number of trials.

Method	Sweep Bean	Insert Screw
π_0	11/16	4/16
MF + RGB Camera	10/16	2/16
MF + Other FFs	-	-
MF + Depth Camera	4/16	7/16
MF + Ours	**14/16**	**15/16**
Method	**Breakfast**	**BiDex Pour**
π_0	4/16	12/16
MF + RGB Camera	5/16	9/16
MF + Other FFs	-	-
MF + Depth Camera	11/16	**16/16**
MF + Ours	**12/16**	**16/16**

TABLE IV: **Results on sim-to-real transfer.** The number of successes over the total number of trials is reported.

Method	Push Cube	Pick Cube
RGB Camera	3/16	2/16
Depth Camera	7/16	5/16
Ours	**16/16**	**12/16**

lation and reality for different methods is presented in Appendix D. RGB images contain redundant information such as high-frequency textures, and the depth in simulation is perfect, whereas real-world depth data are inaccurate and noisy. As a result, directly using RGB, depth, or point cloud data generally leads to a significant sim-to-real gap. Robo3R achieves a consistent scene representation across domains and significantly reduces the sim-to-real gap. In both tasks, Robo3R achieves a notably higher success rate compared to the baselines, as shown in Tab. IV.

3) Grasp Synthesis:

Experimental setup and baselines. We employ AnyGrasp [7] as the grasp synthesis model, using point clouds reconstructed by Robo3R as its input. As baselines, we use point clouds obtained from a depth camera and from the other feed-forward 3D reconstruction models as inputs to AnyGrasp. We conduct experiments on normal, transparent, reflective, and small objects.

Results. The results are presented in Tab. V. The other feed-forward 3D reconstruction models fail to enable AnyGrasp to produce feasible grasps (indicated by "-"). While the depth camera baseline achieved reasonable performance on normal objects (12/16), it struggled with challenging scenarios, dropping to 7/16 for transparent/reflective objects and 6/16 for small objects. In contrast, our method demonstrated superior performance, achieving the highest success rates in all categories. These results confirm that Robo3R significantly improves grasp synthesis robustness, particularly for objects with challenging materials or fine-grained geometries.

4) Collision-Free Motion Planning:

Experimental setup and baselines. We use cuRobo [29] as motion planner. The selection of baselines follows Sec. IV-B3.

Results. Consistent with the grasp synthesis experiments, the other feed-forward 3D reconstruction models failed to enable

TABLE V: **Results on grasp synthesis.** The number of successes over the total number of trials is reported.

Method	Normal	Transparent or Reflective	Small
Other FFs	-	-	-
Depth Camera	12/16	7/16	6/16
Ours	**14/16**	**10/16**	**11/16**

TABLE VI: **Results on collision-free motion planning.** The number of successes over the total number of trials is reported.

Method	Normal	Transparent or Reflective	Thin
Other FFs	-	-	-
Depth Camera	**5/5**	2/5	2/5
Ours	**5/5**	**4/5**	**5/5**

successful obstacle avoidance. While the depth camera proves effective for normal objects, it frequently fails to capture the geometry of challenging obstacles, resulting in a low success rate of 2/5 for transparent, reflective, and thin objects. In contrast, our method maintains high reliability across all scenarios, as shown in Tab. VI. These results highlight that the accurate geometry provided by Robo3R is critical for avoiding collisions with objects that are invisible to depth sensors.

C. Effectiveness of Design Choices

We validate the effectiveness of the proposed extrinsic estimation module (KP + PnP) by comparing it against direct extrinsic prediction (Direct Pred.). As shown in Tab. VII, the extrinsic estimation module yields lower absolute translation and rotation errors, as well as higher absolute translation and rotation accuracy. These results confirm that predicting keypoints followed by a PnP solver provides a more robust and precise estimation of the camera pose compared to directly regressing the extrinsics.

We also investigate the contributions of incorporating robot state during feed-forward reconstruction. The results in Tab. VIII indicate that fusing robot states with image features via element-wise addition not only yields better point map estimation compared to methods that do not use robot states, but also achieves more accurate absolute camera pose in global similarity transformation. Furthermore, this approach outperforms other modality fusion methods, such as employing self-attention within the transformer backbone.

We provide qualitative comparisons for the masked point head, since its advantage in producing finer-grained geometry is not well captured by conventional benchmarks. Specifically, we train a variant that retains the point decoder architecture but removes the masked design (w/o MPH). As shown in Fig. 8, the masked point head enables the model to predict clean points on structures as fine as 1.5 mm, whereas the baseline fails to recover such geometry.

V. Conclusion

In this work, we introduce Robo3R, a feed-forward 3D reconstruction model designed for robotic manipulation. Trained on Robo3R-4M, a large-scale synthetic dataset for perception

TABLE VII: **Ablation study of camera pose prediction in the robot base frame.** We report absolute translation error (ATE), absolute rotation error (ARE), absolute translation accuracy at a threshold of 0.01 meters (ATA@0.01), and absolute rotation accuracy (ARA@0.01).

Method	ATE ↓	ARE ↓	ATA@0.01 ↑	ARA@0.01 ↑
Direct Pred.	0.018	0.018	0.334	0.359
KP + PnP	**0.016**	**0.016**	**0.442**	**0.415**

TABLE VIII: **Ablation study on conditioning on robot state.** The point error, normal error, absolute translation error (ATE), and absolute rotation accuracy (ARA) are reported, with both ATE and ARA evaluated at a threshold of 0.03 meters.

Method	Point Err. ↓	Normal Err. ↓	ATA ↑	ARA ↑
w/o State	0.006	0.081	**0.903**	0.831
Self Attn	0.006	0.086	0.900	0.821
Ours	**0.005**	**0.079**	**0.903**	**0.838**

and reconstruction in robotic manipulation scenarios, Robo3R delivers high-fidelity depth estimation, precise camera parameter prediction, accurate metric scale, and a consistent, canonical coordinate system for 3D representation. Compared to depth cameras, Robo3R is more cost-effective, accurate, robust, and requires no calibration. It reliably handles various object materials and challenging scenarios, significantly enhancing downstream robotic applications such as imitation learning, sim-to-real transfer, grasp synthesis, and collision-free motion planning.

Limitations and future work. Currently, Robo3R supports pinhole cameras and a limited range of embodiment types. Future work may involve generating new data to fine-tune Robo3R, enabling it to adapt to other camera models, such as fisheye and panoramic cameras, as well as an expanded range of embodiments.

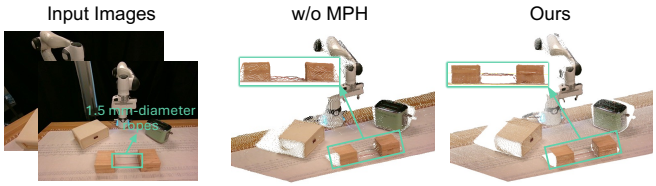

Fig. 8: **Reconstruction w/ and w/o the masked point head.**

References

[1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2024.

[2] Aleksei Bochkovskii, AmaÃĢl Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *International Conference on Learning Representations (ICLR)*, 2025.

[3] Murtaza Dalal, Jiahui Yang, Russell Mendonca, Youssef Khaky, Ruslan Salakhutdinov, and Deepak Pathak. Neural mp: A generalist neural motion planner. In *International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[4] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[5] Siyan Dong, Shuzhe Wang, Shaohui Liu, Lulu Cai, Qingnan Fan, Juho Kannala, and Yanchao Yang. Reloc3r: Large-scale training of relative camera pose regression for generalizable, fast, and accurate visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[6] Zhao Dong, Ka Chen, Zhaoyang Lv, Hong-Xing Yu, Yunzhi Zhang, Cheng Zhang, Yufeng Zhu, Stephen Tian, Zhengqin Li, Geordie Moffatt, et al. Digital twin catalog: A large-scale photorealistic 3d object digital twin dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[7] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 39(5):3929–3945, 2023.

[8] Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J Black, Trevor Darrell, and Angjoo Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[9] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: Infinite resolution action detection transformer for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2023.

[10] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning (CoRL)*, 2023.

[11] Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt-2: Learning precise manipulation from few demonstrations. In *Robotics: Science and Systems (RSS)*, 2024.

[12] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *Conference on Robot Learning (CoRL)*, 2024.

[13] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. *arXiv preprint arXiv:2509.13414*, 2025.

[14] Chung Min Kim, Brent Yi, Hongsuk Choi, Yi Ma, Ken Goldberg, and Angjoo Kanazawa. Pyroki: A modular

toolkit for robot kinematic optimization. in 2025 ieee. In *International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[15] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Ep n p: An accurate o (n) solution to the p n p problem. *International journal of computer vision*, 81 (2):155–166, 2009.

[16] Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. *arXiv preprint arXiv:2510.12276*, 2025.

[17] Puhao Li, Tengyu Liu, Yuyang Li, Yiran Geng, Yixin Zhu, Yaodong Yang, and Siyuan Huang. Gendexgrasp: Generalizable dexterous grasping. In *International Conference on Robotics and Automation (ICRA)*, 2023.

[18] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[19] Tao Lin, Gen Li, Yilei Zhong, Yanwen Zou, Yuxin Du, Jiting Liu, Encheng Gu, and Bo Zhao. Evo-0: Vision-language-action model with implicit spatial understanding. *arXiv preprint arXiv:2507.00416*, 2025.

[20] Minghuan Liu, Zhengbang Zhu, Xiaoshen Han, Peng Hu, Haotong Lin, Xinyao Li, Jingxiao Chen, Jiafeng Xu, Yichu Yang, Yunfeng Lin, et al. Manipulation as in simulation: Enabling accurate geometry perception in robots. *arXiv preprint arXiv:2509.02530*, 2025.

[21] Xinhang Liu, Yuxi Xiao, Donny Y Chen, Jiashi Feng, Yu-Wing Tai, Chi-Keung Tang, and Bingyi Kang. Trace anything: Representing any video in 4d via trajectory fields. *arXiv preprint arXiv:2510.13802*, 2025.

[22] Yiyang Lu, Yufeng Tian, Zhecheng Yuan, Xianbang Wang, Pu Hua, Zhengrong Xue, and Huazhe Xu. H^3dp: Triply-hierarchical diffusion policy for visuomotor learning. *arXiv preprint arXiv:2505.07819*, 2025.

[23] Tyler Ga Wei Lum, Martin Matak, Viktor Makoviychuk, Ankur Handa, Arthur Allshire, Tucker Hermans, Nathan D Ratliff, and Karl Van Wyk. Dextrah-g: Pixels-to-action dexterous arm-hand grasping with geometric fabrics. In *Conference on Robot Learning (CoRL)*, 2024.

[24] Juncheng Mu, Sizhe Yang, Hojin Bae, Feiyu Jia, Qingwei Ben, Boyi Li, Huazhe Xu, and Jiangmiao Pang. One-policy-fits-all: Geometry-aware action latents for cross-embodiment manipulation. *arXiv preprint arXiv:2603.14522*, 2026.

[25] Quanhao Qian, Guoyang Zhao, Gongjie Zhang, Jiuniu Wang, Ran Xu, Junlong Gao, and Deli Zhao. Gp3: A 3d geometry-aware policy with multi-view images for robotic manipulation. *arXiv preprint arXiv:2509.15733*, 2025.

[26] Yuzhe Qin, Binghao Huang, Zhao-Heng Yin, Hao Su, and Xiaolong Wang. Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation. In *Conference on Robot Learning (CoRL)*, 2023.

[27] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2023.

[28] Edgar Sucar, Zihang Lai, Eldar Insafutdinov, and Andrea Vedaldi. Dynamic point maps: A versatile representation for dynamic 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[29] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, et al. Curobo: Parallelized collision-free robot motion generation. In *International Conference on Robotics and Automation (ICRA)*, 2023.

[30] Bin Tan, Changjiang Sun, Xiage Qin, Hanat Adai, Zelin Fu, Tianxiang Zhou, Han Zhang, Yinghao Xu, Xing Zhu, Yujun Shen, et al. Masked depth modeling for spatial perception. *arXiv preprint arXiv:2601.17895*, 2026.

[31] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[32] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[33] 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 *International Conference on Robotics and Automation (ICRA)*, 2023.

[34] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[35] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[36] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[37] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. pi^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025.

[38] Zhenyu Wei, Zhixuan Xu, Jingxiang Guo, Yiwen Hou, Chongkai Gao, Zhehao Cai, Jiayu Luo, and Lin Shao. D (r, o) grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping. In *International Conference on Robotics and Automation (ICRA)*, 2025.

[39] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. Foundationstereo: Zero-shot stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[40] Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, Tsung-Wei Ke, and Katerina Fragkiadaki. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2023.

[41] Gangwei Xu, Haotong Lin, Hongcheng Luo, Xianqi Wang, Jingfeng Yao, Lianghui Zhu, Yuechuan Pu, Cheng Chi, Haiyang Sun, Bing Wang, et al. Pixel-perfect depth with semantics-prompted diffusion transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[42] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[43] Ge Yan, Yueh-Hua Wu, and Xiaolong Wang. Dnact: Diffusion guided multi-task 3d policy learning. In *International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[44] Ge Yan, Jiyue Zhu, Yuquan Deng, Shiqi Yang, Ri-Zhao Qiu, Xuxin Cheng, Marius Memmel, Ranjay Krishna, Ankit Goyal, Xiaolong Wang, et al. Maniflow: A general robot manipulation policy via consistency flow training. In *Conference on Robot Learning (CoRL)*, 2025.

[45] Jiahui Yang, Jason Jingzhou Liu, Yulong Li, Youssef Khaky, Kenneth Shaw, and Deepak Pathak. Deep reactive policy: Learning reactive manipulator motion planning for dynamic environments. In *Conference on Robot Learning (CoRL)*, 2025.

[46] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[47] Sizhe Yang, Yiman Xie, Zhixuan Liang, Yang Tian, Jia Zeng, Dahua Lin, and Jiangmiao Pang. Ultradexgrasp: Learning universal dexterous grasping for bimanual robots with synthetic data. *arXiv preprint arXiv:2603.05312*, 2026.

[48] Zhecheng Yuan, Tianming Wei, Shuiqi Cheng, Gu Zhang, Yuanpei Chen, and Huazhe Xu. Learning to manipulate anywhere: A visual generalizable framework for reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2024.

[49] Zhecheng Yuan, Tianming Wei, Langzhe Gu, Pu Hua, Tianhai Liang, Yuanpei Chen, and Huazhe Xu. Hermes: Human-to-robot embodied learning from multi-source motion data for mobile dexterous manipulation. *arXiv preprint arXiv:2508.20085*, 2025.

[50] Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. In *Conference on robot learning (CoRL)*, 2023.

[51] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Robotics: Science and Systems (RSS)*, 2024.

[52] Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K Barral, Raia Hadsell, et al. Efficiently reconstructing dynamic scenes one d4rt at a time. *arXiv preprint arXiv:2512.08924*, 2025.

[53] Chuye Zhang, Xiaoxiong Zhang, Wei Pan, Linfang Zheng, and Wei Zhang. Generative visual foresight meets task-agnostic pose estimation in robotic table-top manipulation. In *Conference on robot learning (CoRL)*, 2025.

[54] Jialiang Zhang, Haoran Liu, Danshi Li, XinQiang Yu, Haoran Geng, Yufei Ding, Jiayi Chen, and He Wang. Dexgraspnet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes. In *Conference on Robot Learning (CoRL)*, 2024.

[55] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. In *International Conference on Learning Representations (ICLR)*, 2025.