RoST3R: Robot-Aware Dynamic 3D Reconstruction from RGB Input for Robotic Manipulation

Author Names Omitted for Anonymous Review.

***Abstract*— 3D scene representations offer stronger generalization for policy learning compared to 2D representations, yet collecting such 3D data has required special sensors. Previous methods for 3D reconstruction from video exist, but have been unsuitable for robotic learning due to error and lack of metric calibration. In this work, we demonstrate that 3D scene representations can be reliably reconstructed from standard 2D RGB images, making it both accessible and practical for robot learning. We propose a novel framework, RoST3R (Robot MonST3R), that incrementally reconstructs dynamic 3D scenes at metric scale from RGB images, enabling 3D-aware policy learning in complex environments from only 2D inputs. At its core, our approach estimates the robot's pose during scene reconstruction, registers its kinematic structure within the environment, and builds a unified 3D scene representation. This unified 3D representation offers two key benefits: it enables policy learning at metric scale in a consistent world frame—decoupling object and camera dynamics—and provides a coherent model of the robot and environment to support fine-grained spatial reasoning. Notably, while the input remains 2D, our approach generates a 3D-aware representation that significantly improves generalization. Experiments show that policies trained with this 3D representation outperform those trained on 2D inputs, particularly in tasks involving environmental variations, novel viewpoints and camera motion. In simulation, our method outperforms 2D counterparts by 24.5% under environmental variations and dynamic camera motion. In real-world scenarios, it achieves a 29.5% performance improvement. Video results are available on the anonymous webpage: https://rost3r-project.github.io/.**

I. INTRODUCTION

Learning generalizable robot policies requires a stable and structured understanding of the surrounding environment. RGB images serve as the primary modality for many contemporary visuomotor policy learning approaches [1], [2], yet it is well known that representations based on 2D observations struggle with generalization across varying viewpoints, object arrangements, and dynamic environments. In contrast, 3D representations offer increased invariance to perspective and object configuration, making them an attractive foundation for robust policy learning. 3D representations (e.g., point clouds) have long been used in robotics, but typically require dedicated sensors. Reconstructing 3D scenes from monocular video has been a long-standing goal in computer vision and robotics, e.g., SLAM [3], [4], but until recently, reconstruction methods have not achieved the level of accuracy and density needed for common robotic tasks.

In this work, we propose a novel framework that reconstructs dynamic 3D scenes directly from standard 2D RGB inputs, making 3D-aware policy learning broadly accessible without the need for specialized sensors. Our key insight is that a robot's known kinematic structure can serve as a reliable anchor within the reconstructed scene, enabling metrically accurate scaling and consistent scene registration even from partial and noisy observations. By incrementally estimating the robot pose during scene reconstruction and embedding the robot within the reconstructed scene, our framework builds a unified 3D scene representation that evolves over time, even under camera motion.

This unified 3D representation brings two critical advantages. First, it enables policy learning at metric scale in a consistent world-coordinate frame, effectively decoupling object dynamics from camera motions and alleviating the burden of viewpoint variation. Second, it ensures that both the robot and the environment are coherently modeled, supporting interactions that require fine-grained spatial reasoning. Importantly, while the system relies only on 2D inputs, the resulting 3D-aware policies achieve significantly stronger generalization compared to policies trained on raw image observations.

We validate our approach across a range of manipulation tasks with varying complexity, including dynamic environments with moving camera. Experimental results show that policies trained using our 3D scene reconstruction framework consistently outperform 2D-based baselines—by 24.5% in simulation, and by 29.5% in real-world scenarios—particularly in challenging settings with environmental variations, novel viewpoints, and camera motion. By bridging the gap between 2D perception and 3D-aware policy learning, our work opens new opportunities for robust and scalable robot learning in complex real-world settings.

II. RELATED WORK

Dynamic 3D Reconstruction. Recovering accurate camera parameters and consistent dense geometry from monocular videos of dynamic scenes is an important yet highly challenging task in computer vision. Classical structure-from-motion (SfM) [5], [6], [7], [8] and simultaneous localization-and-mapping (SLAM) [9], [3], [10], [11], [12] pipelines laid the algorithmic groundwork for rigid, static worlds, but their scale ambiguities and static-scene assumptions limit performance under real-world motion. Recent methods such as Robust-CVD [13] and CasualSAM [14] address this by jointly estimating camera parameters and dense depth maps from dynamic videos, either through optimizing a spatially varying spline or by fine-tuning monocular depth networks. MegaSaM [4] extends deep visual SLAM frameworks [3] with new training and inference schemes to handle dynamic scenes more effectively. Meanwhile, methods like MonST3R

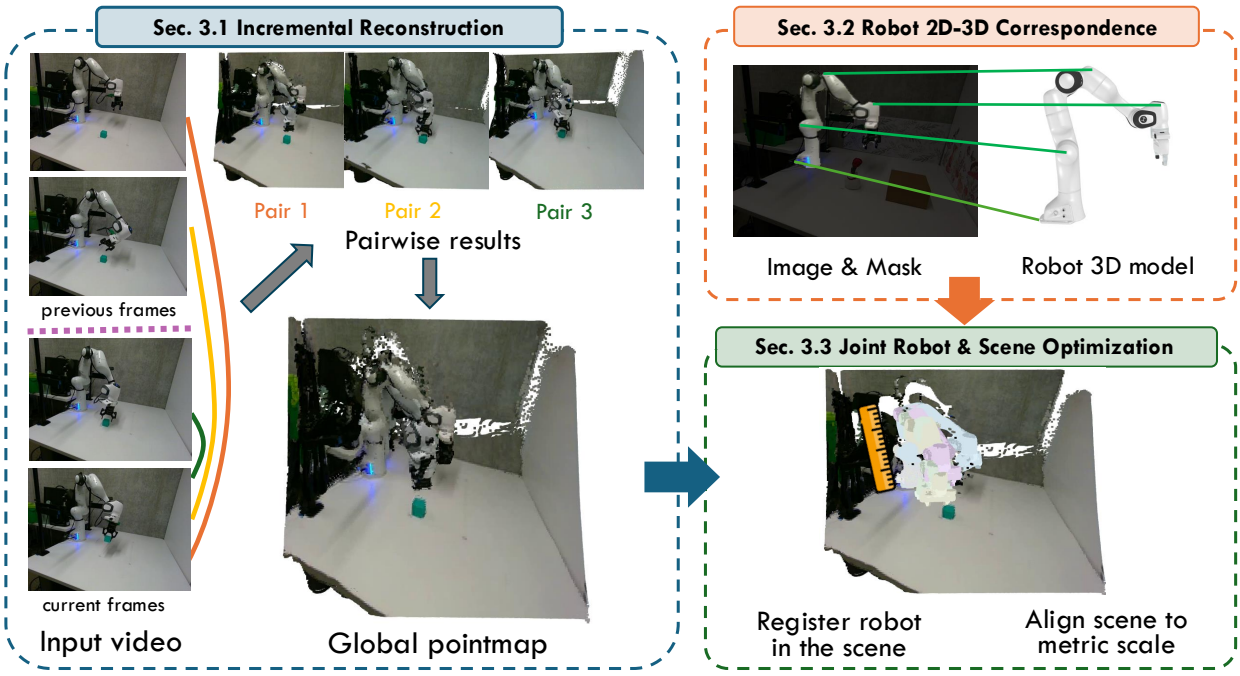

Fig. 1: **Method overview. RoST3R** extends MonST3R for incremental dynamic 3D reconstruction in the world coordinate, by adjusting the pair-sampling strategy for a global streaming pointmap optimization (Section III-A). Then, by aligning the robot 3D model with 2D observations in each frame (Section III-B), RoST3R can reliably register the robot into the environment in a unified 3D space, as well as calibrate the environment reconstruction to be metric-scale (Section III-C).

[15], Align3R [16], CUT3R [17] and MASt3R [18] adopt a 3D point map representation from DUSt3R [19] and localize the camera either through optimization-based or feedforward strategies. Our method builds on this trajectory with key improvements over MonST3R that are critical for robotics: (i) it supports streaming input through incremental optimization, while MonST3R requires batch processing; (ii) it recovers metric scale by registering the robot's full kinematic structure into the reconstructed scene; and (iii) it fine-tunes the visual backbone on synthetic data to mitigate distribution shifts in robotic scenes.

Robotic Manipulation in 2D and 3D. 2D camera images have been widely adopted for vision-based robotic manipulation. Methods like [1], [20], [21], [2], [22] employ end-to-end image-to-action models for policy learning. However, these approaches often struggle in tasks that require high precision, robust spatial interactions, or resilience to environmental and camera variations. This is largely because 2D images inherently lack depth information, making it difficult to accurately perceive object geometry, spatial relationships, and subtle environmental changes. Recent advances have addressed these limitations by incorporating 3D perception. 3D based methods [23], [24], [25], [26], [27] have demonstrated stronger generalization capabilities compared to their 2D counterparts. Nevertheless, in real-world deployment, these 3D methods rely on depth sensors, which are prone to sensor limitations and noise. In contrast, our method develops a 3D representation directly from 2D images, eliminating the need for a depth sensor while achieving strong generalization performance.

III. RoST3R: Robot-Aware Dynamic 3D Reconstruction

We introduce RoST3R, a framework for reconstructing robot-aware, scale-aligned 3D scene representations directly from RGB images to boost robot manipulation policy learning. Our framework employs a fine-tuned pointmaps prediction backbone to estimate contact-aware 3D pointmaps from RGB-only input. These pairwise pointmaps are then incrementally fused to represent the scene and objects in a unified manner (Section III-A). Alongside scenes and objects, robots as the main contact agents play a critical role in policy performance. RoST3R establishes precise correspondences between robot's 2D projection on the input images and robot's 3D kinematic structure to enable accurate 6DoF robot pose estimation during optimization (Section III-B). Finally, by enforcing robot-aware constraints using these correspondences, RoST3R registers the robot in the scene by optimizing the robot pose during the incremental reconstruction (Section III-A) and recovers a globally consistent, metrically scaled reconstruction (Section III-C). The entire framework is illustrated in Figure 1.

Problem Formulation. Given a sequence of N RGB images $\{\mathbf{I}_t\}_{t=1}^{N}$, each image captures both the robot and its surrounding environment under camera motion, along with the corresponding robot joint configurations $\{\mathbf{q}_t\}_{t=1}^{N}$, where $\mathbf{q}_t$ denotes the joint positions at time t, our goal is to incrementally reconstruct both the robot and the environment into a robot-aware, scale-aligned 3D representation. For a single

image $\mathbf{I}_t$, RoST3R predicts a pointmap $\mathbf{X}_t \in \mathbb{R}^{H\times W\times 3}$ in global frame, estimates the transformation $\mathbf{P}_t^{\text{robot}} \in \text{SE}(3)$ of robot that aligns the robot geometry (defined by joint configuration $\mathbf{q}_t$) to global frame, and infers a global scale factor σ to recover the metric scale of the scene simultaneously. These outputs are incrementally fused to build unified 3D pointmaps of the robot and its environment over time.

A. Reconstructing Scene Representation

Preliminaries. Given a pair of images captured at times t and t', methods such as DUSt3R [19] and MonST3R [15] employ a ViT-based network [28] to predict two corresponding pointmaps, $\hat{\mathbf{X}}_t$ and $\hat{\mathbf{X}}_{t'}$. These pointmaps represent the 3D structure of images t and t', respectively, and are both expressed in the coordinate frame of the camera at time t. The hat symbol indicates that the prediction is expressed in the local coordinate frame of the t-th image, while the subscript denotes the image to which the pointmap corresponds. In contrast, symbols without the hat (e.g., $\mathbf{X}_t$) represent pointmaps that have been transformed into a global or world coordinate frame. For example, $\hat{\mathbf{X}}_{t'}$ denotes the pointmap at time t' in the coordinate frame of image t, while $\mathbf{X}_t$ refers to the globally aligned pointmap of image t.

Each 3D pointmap $\mathbf{X}_t$ is recovered by re-parameterizing the scene using the camera extrinsics $\mathbf{P}_t = [\mathbf{R}_t \,|\, \mathbf{T}_t]$, intrinsics $\mathbf{K}_t$, and the corresponding per-frame depth map $\mathbf{D}_t$. Each 3D point in the pointmap is computed as: $\mathbf{x}_t^{i,j} := \mathbf{P}_t^{-1}\, h(\mathbf{K}_t^{-1}[i\mathbf{D}_t^{i,j},\ j\mathbf{D}_t^{i,j},\ \mathbf{D}_t^{i,j}]^T)$, where (i, j) denotes the pixel coordinates and $h(\cdot)$ represents the transformation to homogeneous coordinates. Thus, $\mathbf{X}_t$ is a reparameterized form of $(\mathbf{P}_t, \mathbf{K}_t, \mathbf{D}_t)$ in the global frame.

Streaming Pointmaps Estimation. MonST3R [15] constructs a dense connectivity graph by sampling all frame pairs within a global sliding window. However, this approach assumes access to the entire dataset in advance and incurs a quadratic time complexity, making it impractical for long sequences due to computational and memory overhead. To address these limitations, we propose a sequential and incremental graph construction strategy designed for streaming robot observations, where frames arrive over time and future data is unavailable.

Let the robot's visual observations be denoted as $\mathbf{O} = \{\mathbf{I}_t\}_{t=1}^{N}$, with each frame $\mathbf{I}_t$ captured at time t. At incremental reconstrcution step k, a batch of l new frames $\mathbf{O}' = \{\mathbf{I}_{N+1}, \ldots, \mathbf{I}_{N+l}\}$ is received, where the batch size $l \geq 1$. We maintain a temporal window of size w, and only form observation pairs (i, j) where the temporal offset between frames satisfies $0 < j - i \leq w$.

We construct edges under the following rules:

1) Forward edges from fixed to new observations: Previously optimized frames $\mathbf{I}_i \in \mathbf{O}$ connect to new frames $\mathbf{I}_j \in \mathbf{O}'$ only if $j - i \leq w$:

$$\mathbf{W}_k^{\text{old-to-new}} = \{(i, j) \mid i \in \mathbf{O},\, j \in \mathbf{O}',\, 0 < j - i \leq w\}$$

2) Internal edges within the new batch: New observations $\mathbf{I}_i, \mathbf{I}_j \in \mathbf{O}'$ also form pairs, but still obey the same sliding window constraint:

$$\mathbf{W}_k^{\text{new-to-new}} = \{(i, j) \mid i, j \in \mathbf{O}',\, 0 < |j - i| \leq w\}$$

The total set of edges at step k is $\mathbf{W}_k = \mathbf{W}_k^{\text{old-to-new}} \cup \mathbf{W}_k^{\text{new-to-new}}$. During this step, only the new observations in $\mathbf{O}'$ are optimized, while all earlier frames in $\mathbf{O}$ remain unchanged.

Global Alignment. With the sampled frame pair, we now detail the optimization process. Our goal is to align all local-frame pointmap pairs (e.g., $\hat{\mathbf{X}}_t$, $\hat{\mathbf{X}}_{t'}$) into a unified pointmap $\mathbf{X}_t \in \mathbb{R}^{H\times W\times 3}$ in the global frame. We begin by adapting the alignment term from MonST3R [15], which optimizes the rigid transformations $\{\mathbf{P}_e\}_{e\in\mathbf{W}}$ for each pair of pointmaps, aligning them to the global frame:

$$\mathcal{L}_{\text{align}} = \sum_{e\in\mathbf{W}} \sum_{t\in e} \left\| \mathbf{C}_t \cdot \left(\mathbf{X}_t - s_e \mathbf{P}_e \hat{\mathbf{X}}_t \right) \right\|_1 \tag{1}$$

Here, s_e is a pairwise scale factor, $\mathbf{C}_t$ is the confidence map and we optimize $\mathbf{X}_t$ only when $t \in \mathbf{O}'$. To ensure smooth transitions between the last frame of the previous step and the first frame of the current step, we modify the standard camera trajectory smoothness loss from [15]. We introduce a weighting factor β to emphasize the transition between old and new frames:

$$\begin{aligned}\mathcal{L}_{\text{smooth}} = \beta \left(\left\| \mathbf{R}_N^\top \mathbf{R}_{N+1} - \mathbf{I} \right\|_F + \left\| \mathbf{T}_{N+1} - \mathbf{T}_N \right\|_2 \right) \\ + \sum_{t\geq N+1} \left\| \mathbf{R}_t^\top \mathbf{R}_{t+1} - \mathbf{I} \right\|_F + \left\| \mathbf{T}_{t+1} - \mathbf{T}_t \right\|_2\end{aligned} \tag{2}$$

where t_N denotes the final image of the previous batch, t_{N+1} is the first image of the current batch, and the Frobenius norm $\|\cdot\|_F$ is used to measure the rotation difference.

Finetuning for Contact-Aware Dynamic 3D Estimation. While MonST3R [15] primarily focuses on reconstructing dynamic scenes in daily life, the proposed RoST3R is tailored for reconstructing scenes involving robotic manipulation, which introduces a distribution shift. A notable limitation of MonST3R is its inability to reconstruct the contact region between the robot and the object it is intended to interact with. This challenge motivates the fine-tuning of the model to better suit the specific requirements of robotic manipulation.

To address this gap, we first curate a synthetic robotic dataset derived from RoboVerse [29], which includes a variety of manipulation tasks performed by a simulated Franka robot, and then fine-tune the depth conditioned variant [16] of MonST3R [15] based on it. These tasks encompass picking and stacking cubes, opening and closing containers, and interacting with articulated objects such as closets and drawers, etc. RoboVerse provides high-fidelity synthetic data with ground truth annotations for depth and camera pose, enabling accurate supervision for dynamic geometry estimation. The dataset captures realistic object dynamics and fine-grained contact interactions, which are crucial for 3D understanding in robotics. To enhance generalization, it also includes environment, camera, and lighting/reflection randomizations. Training details are provided in the Sec IV-A.

B. Establishing Correspondence

Integrating the robot's kinematic structure into the reconstruction process relies on establishing accurate correspondences between 2D images and the robot's 3D geometry. These correspondences enable robust estimation of the robot pose during optimization and its alignment within the reconstructed scene.

Grounding Robot in Images. We use Grounded SAM [30] to perform robot segmentation. We first employ Grounding DINO [31] to detect the robot in the initial frame of the video. Subsequently, we perform video segmentation using the SAM-2 model [32]. Specifically, we segment objects in the first frame based on the bounding boxes obtained from detection and select five positive pixel points within each object mask to guide the segmentation. This process produces robot segmentation masks for each frame. Using these segmentation masks $\{\mathbf{M}_t\}_{t=1}^{N}$, we then estimate the proxy robot pose $\mathbf{P}_t^{\text{fp}}$ in each frame, where proxy denotes that $\mathbf{P}_t^{\text{fp}}$ is used solely to establish 2D–3D correspondences between the image and the robot's 3D geometry.

Aligning Robot with 2D Segmentations. We build on model-based FoundationPose [33], which estimates the 6-DoF pose $\mathbf{P}_t^{\text{fp}}$ of the robot using the robot mesh, camera intrinsics, an RGB image, and a corresponding depth map. However, since our framework relies solely on RGB images to estimate depth, FoundationPose cannot be directly applied in its original form.

To address this, we introduce a novel approach that uses a monocular depth estimation model to predict depth and camera intrinsics directly from RGB images. We also propose a scale alignment strategy between the predicted depth and the robot mesh, enabling accurate pose estimation without requiring ground-truth depth information.

Specifically, we utilize the intrinsic and depth map estimated for depth conditioned pointmaps prediction [16] , and align the scale between the predicted depth and the robot mesh. The scale ratio is computed as the ratio of the maximum distances between points in the unprojected RGB depth map and the robot mesh, denoted as $\rho = \frac{l_{\text{image}}}{l_{\text{mesh}}}$, where l_{image} is the maximum Euclidean distance between unprojected points in the camera frame, and l_{mesh} is the maximum distance between vertices in the robot mesh. However, this estimated scale may be affected by inaccuracies in depth prediction, camera intrinsic estimation, and partial occlusions. To mitigate such errors, we introduce a scale selection procedure during pose estimation for the first frame. We define a maximum scaling factor α and perform a search over a scale range $[\frac{\rho}{\alpha}, \alpha\rho]$ with a fixed step size.

To select the optimal scale, we minimize the intersection-over-union (IoU) loss between the projected robot mask and the ground-truth mask. The IoU measures the overlap between the predicted and true masks, and we seek to maximize this overlap by adjusting the scale. To improve efficiency, we discard candidate scales where the projected mesh covers too few pixels or extends beyond the image boundaries.

C. Robot Structure Guided Scale Alignment

To register the robot into the scene and recover the metric scale, we introduce a dense 3D-3D alignment loss, which is based on the 2D-3D correspondences derived in Section III-B. Let $\mathbf{X}_t^{\text{robot}} \in \mathbb{R}^{H\times W\times 3}$ be the per-pixel canonical robot points corresponding to $\mathbf{X}_t$, obtained via 2D-3D correspondences. We estimate a transformation $\mathbf{P}_t^{\text{robot}} \in \text{SE}(3)$ that aligns the robot mesh with the world frame, and jointly optimize $\mathbf{P}_t^{\text{robot}}$ along with the predicted geometry and the global scale factor σ to enforce metric consistency. The robot depth loss is defined as:

$$\mathcal{L}_{depth} = \sum_{t\in\mathbf{O}'} \|\mathbf{M}_t\mathbf{C}_t(\sigma\mathbf{X}_t - \mathbf{P}_t^{\text{robot}}\mathbf{X}_t^{\text{robot}})\|_2 \qquad (3)$$

where $\mathbf{M}_t$ is the segmentation mask obtained from object grounding (Section III-B). The complete optimization for estimating the dynamic, metrically scaled global point cloud and camera poses is formulated as:

$$\{\mathbf{X}\} = \underset{\{\mathbf{X},\mathbf{P},s,\sigma\}}{\arg\min}\ \mathcal{L}_{\text{align}} + w_{\text{depth}}\mathcal{L}_{\text{depth}} + w_{\text{smooth}}\mathcal{L}_{\text{smooth}} \quad (4)$$

IV. EVALUATION

A. Implementation Details

Training Detail. We finetune our model using four RTX 6000 Ada GPUs with a batch size of 5 for approximately 50 epochs. The AdamW optimizer is employed with a learning rate of 0.00005. Input images are randomly resized to one of three resolutions: 512×288, 512×336, or 512×256. Each epoch comprises 25,000 image pairs sampled from 6,000 episodes in [29].

Synthetic Dataset. We construct a synthetic dataset by selecting 120 tasks from [29]. For each task, we sample 50 episodes with systematic variation in object textures, camera pose, illumination (including reflections), and camera motion. We retain both successful and failed rollouts to improve RoST3R's generalization ability.

Optimization Detail. We set the temporal window size w to 4 and the maximum scaling factor α for pose estimation to 5.0. The weighting factor β in the camera trajectory smoothness loss (Eq. 2) is set to 3.0. For the global optimization (Eq. 4), we use $w_{\text{depth}} = 0.5$ and $w_{\text{smooth}} = 0.01$ as the weights for the depth and smoothness terms, respectively. Optimization is performed using the Adam optimizer for 40 iterations with a learning rate of 0.01.

B. 6D Pose Estimation with Known Joint Angles

We evaluate our pose estimation method (Section III-B) on a practical task where an image of a robot with known, fixed joint angles is used to estimate the 6D camera-to-robot pose.

Evaluation Protocol. We evaluate our method on the Panda-3CAM dataset, following prior works [34], [35]. This real world dataset captures a Panda robot performing various motions, recorded by three fixed cameras with different focal lengths and resolutions. We use a 2D keypoint-based ADD metric [34], which assesses pose estimation accuracy by computing the average distance between corresponding

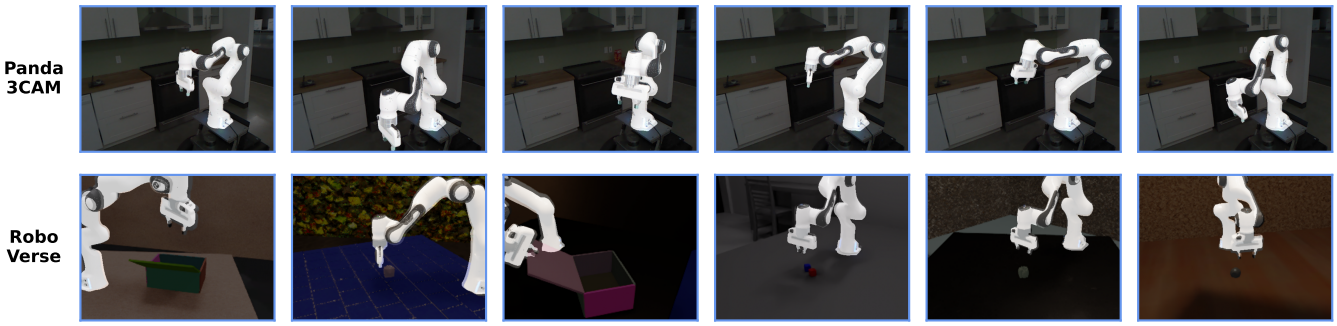

Fig. 2: **Visualization of pose estimation.** We visualize the results of projecting the robot mesh onto the image using the estimated pose from our method. The first row shows data from the real-world Panda-3CAM dataset, while the second row shows data from the synthetic RoboVerse [29] dataset with partial robot visibility.

keypoints at the robot's joint locations. These keypoints are projected into the image plane by transforming the 3D joint locations into the camera frame using both the predicted and ground-truth poses, followed by applying the camera intrinsics. We report the AUC (area under curve) of the ADD metric across different error threshold.

Baselines. We compare our method against three methods: DREAM [34], RoboPose [35], and the original FoundationPose [33]. DREAM uses a deep neural network to localize a fixed set of predefined keypoints in the image, followed by a 2D-to-3D optimization to recover the robot's 6D pose. RoboPose estimates the pose from a single image by iteratively refining the rendering of the robot and predicting delta joint angles relative to the rendered CAD model.

Results. As shown in Table I, our method achieves state-of-the-art results on two of the three datasets. In contrast, FoundationPose fails entirely due to the absence of scale adjustment. Notably, both DREAM and RoboPose require per-object training, whereas our method generalizes without any additional training. As illustrated in Figure 2, our estimations remain accurate and robust, even under severe occlusions.

TABLE I: **Pose estimation quality**. We report AUC (higher is better) metrics for different methods. *: It is worth noting that our method, performing zero-shot evaluation, achieves competitive and even better results than other baselines that require specific training. FoundationPose (FP) completely fails on pose estimation due to scale misalignment.

Dataset	DREAM [34]	RoboPose [35]	FP [33]	**Ours***
Kinect360	76.26	78.28	0.00	**80.09**
Realsense	81.27	76.44	0.00	**81.47**
Azure	76.54	**80.03**	0.00	69.37

C. Evaluation on RoboVerse Simulation

We evaluate the generalization ability of our proposed 3D representation, RoST3R, on the imitation learning benchmark provided by RoboVerse [29].

Evaluation Protocol. To rigorously evaluate policy generalization, we select one representative task from each of the four source benchmarks included in RoboVerse [29]: CloseBox (RLBench [36]), PickCube (ManiSkill [37]), MoveSliderLeft (CALVIN [38]), and PickChocolatePudding (LIBERO [39]). RoboVerse defines four levels of generalization, each introducing increased complexity while preserving variations from the preceding levels. To further challenge 3D reasoning capabilities, we introduce an additional randomization (Level 4), where the camera follows a random trajectory within a single sequence, simulating dynamic viewpoint changes (Table II).

We comprehensively evaluate our models on all of the Levels. All policies are trained on 50 demonstrations and evaluated on 30 unseen demonstrations, with task completion rate as the evaluation metric.

Baselines. To fairly assess RoST3R's generalization, we compare it to two state-of-the-art imitation learning baselines that also rely solely on RGB input: Diffusion Policy [1] and ACT [20]. DP3 [23] is a 3D extension of Diffusion Policy that conditions on point clouds generated by depth sensors. In our setup, we adopt DP3 as the imitation learning algorithm and condition it on point clouds reconstructed from RGB images using RoST3R. We report evaluation results for both RoST3R-DP3, which uses RoST3R reconstructed point clouds, and Oracle-DP3, which utilizes ground-truth point clouds. We include Oracle-DP3 as an upper-bound reference.

Results. We present quantitative results for the five levels in Table III. At Levels 0 and 1, the tasks are relatively straightforward, with minimal reliance on complex 3D reasoning. In these early stages, our method performs below the 2D-based baselines but remains competitive overall. However, as the difficulty of the tasks increases, the advantages of RoST3R's 3D representation become increasingly pronounced. By Level 3—where variations in both viewpoint and lighting are introduced—our method demonstrates a notable performance gain, surpassing ACT [20] by 9.8%. This improvement underscores the strength of the spatial reasoning capabilities afforded by our 3D approach. At Level 4, the challenge intensifies further with the inclusion of dynamic camera motion. In this scenario, RoST3R signeficantly outperforms ACT by **24.5%**. This result highlights the critical importance of maintaining a consistent and coherent global 3D representation when navigating environments with high visual and geometric variability.

TABLE II: **Generalization levels of evaluation on RoboVerse benchmark** [29].

Level	Description
0	Task-space generalization with a fixed environment, camera, and lighting.
1	Level0 + environment randomization (e.g., object textures, distractors).
2	Level1 + camera randomization.
3	Level2 + lighting/reflection randomization.
4	Level3 + camera motion for a single sequence.

TABLE III: **Task completion rate on RoboVerse benchmark**

Method	CloseBox	PickCube	MoveSliderLeft	PickChocolatePudding	Average	
Diffusion Policy [1]	90.00	33.33	86.67	46.67	64.17	Level 0
ACT [20]	76.67	76.67	86.67	100.0	**85.00**	
Oracle-DP3 [23]	96.67	33.33	83.33	100.0	78.33	
RoST3R-DP3	66.67	26.67	80.00	100.0	68.34	
Diffusion Policy [1]	53.33	13.33	63.33	26.67	39.17	Level 1
ACT [20]	50.00	20.00	83.33	100.0	**63.33**	
Oracle-DP3 [23]	56.67	30.00	73.33	83.33	60.83	
RoST3R-DP3	53.33	20.00	73.33	83.33	57.50	
Diffusion Policy [1]	43.33	6.67	63.33	26.67	35.00	Level 2
ACT [20]	30.00	6.67	83.33	100.0	55.00	
Oracle-DP3 [23]	66.67	16.67	80.00	100.0	65.84	
RoST3R-DP3	53.33	16.67	76.67	76.67	**55.84**	
Diffusion Policy [1]	10.00	3.33	63.33	46.67	30.83	Level 3
ACT [20]	20.00	6.67	80.00	96.67	50.84	
Oracle-DP3 [23]	46.67	13.33	80.00	100.0	60.00	
RoST3R-DP3	46.67	10.00	80.00	86.67	**55.84**	
Diffusion Policy [1]	10.00	3.33	60.00	43.33	29.17	Level 4
ACT [20]	23.33	6.67	70.00	63.33	40.83	
Oracle-DP3 [23]	36.67	13.33	76.67	90.00	54.17	
RoST3R-DP3	40.00	10.00	70.00	83.33	**50.83**	

D. Evaluation in the Real World

For real-world evaluation, we design five tasks of different complexity: Pick Cube, Put Plant into Basket , Place Rose on Box, Water the Plant, and Pour Corn. (see Figure 3). These tasks are selected to cover a range of skills, including precise grasping, object relocation, and tool-use-like motions, thereby assessing the generality and robustness of our policy. For each task, we collect 40 demonstrations using 3D mouse teleoperation.

Hardware Setup. For our real-world experiments, we use a Franka Emika Panda robotic arm. The robot is controlled via a high-level interface that integrates seamlessly with our planning and control pipeline.

Evaluation Protocol. We evaluate policy performance using the task completion rate. For each task and each policy, we conduct 10 rollouts and compute the mean completion score across these trials.

Baselines. We train both our proposed method and all baseline approaches following the same protocol used in the simulation benchmark. We evaluate our method and compare it against the baseline Diffusion Policy [1].

Results. As shown in Table IV, our method outperforms Diffusion Policy by a substantial margin of **29.5%** in average task completion rate. This performance gain highlights our method's enhanced 3D spatial reasoning capabilities and superior generalization across diverse task configurations, proving the necessity of using 3D representations for effective policy learning. Moreover, our method recovers from errors more effectively than Diffusion Policy, adapting and re-planning when facing unexpected failures.

TABLE IV: **Task completion rate on real-world tasks.**

Task	**DP [1]**	**RoST3R-DP3**
Pick Cube	0.30	**0.50**
Put Plant into Basket	0.15	**0.25**
Place Rose on Box	0.55	**0.70**
Water the Plant	0.40	**0.50**
Pour Corn	0.80	**0.90**
Average	0.44	**0.57**

E. Ablation Study

Backbone. To evaluate the effectiveness of different backbones in estimating accurate pairwise pointmaps, we conduct an ablation study comparing our proposed backbone with MonST3R [15] and Align3R [16]. We sample 1,000 image pairs from 20 unseen diverse scenes in the RoboVerse [29] dataset. For each method, we compute the Chamfer Distance between the normalized predicted point clouds and the normalized ground truth, using it as the evaluation metric.

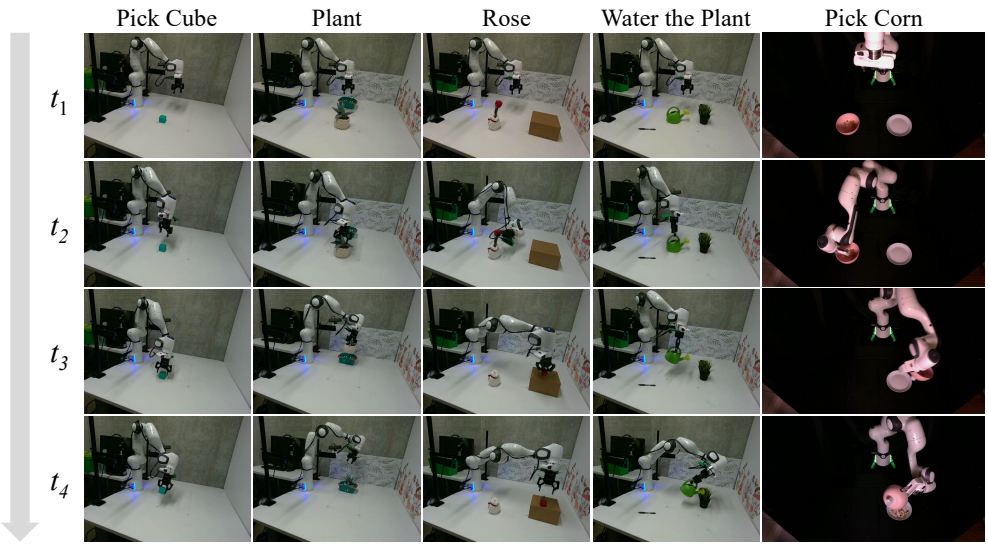

Fig. 3: **Real world evaluation of RoST3R on different tasks.**

The results are presented in Fig.4. Our method achieves a significantly lower Chamfer Distance, indicating more accurate point cloud predictions. We attribute this improvement to the contact-aware fine-tuning. described in Sec.III-A, which enhances the model's ability to reason about object geometry in manipulation scenarios.

Incremental Alignment. We conduct an ablation study on the MoveSliderLeft task [38] at Level 0 to evaluate the impact of key design choices in our incremental alignment optimization. Specifically, we compare our full method with two ablated variants: (1) *w/o metric scale*, where the reconstructed point cloud is not scaled metrically, and (2) *w/o full robot*, where only a partial robot geometry is registered into the scene. Figure 5 presents the success rates of the different variants, emphasizing the critical role of both metric scaling and complete robot registration for effective spatial reasoning and high task performance.

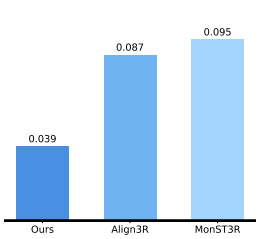

Fig. 4: **Chamfer distances for different methods.**

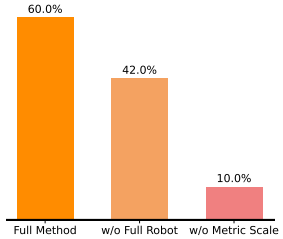

Fig. 5: **Success rates on the MoveSliderLeft task (level 0).**

Running Time. RoST3R operates at approximately 1 frame per second (FPS) to generate a 3D representation from a single image. We provide a breakdown of the runtime for each component of our method in Table V. During the Mesh Extraction stage, we obtain the template robot mesh used by FoundationPose [33]. This mesh is then used to generate a point cloud, which serves as input for computing the depth loss term $\mathcal{L}_{\text{depth}}$ in the Optimization stage. Note that segmentation is performed using Grounded-SAM in parallel with mesh extraction. Since it incurs less computational cost than mesh extraction, we omit its runtime from the table. The Pose Tracking stage estimates $\mathbf{P}^{fd}$ to establish correspondences between 2D image and the 3D robot model. Finally, in the Optimization stage, we reconstruct the dynamic 3D scene and estimate the robot's pose $\mathbf{P}^{\text{robot}}$ for metric-scale recovery.

TABLE V: **Running time analysis.**

Stage	Time (s)
Mesh Extraction	0.32
Point Cloud Generation	0.03
Pose Tracking	0.18
Optimization	0.44
Total Time	0.97

Computation Memory. To demonstrate the memory efficiency of our method, we conduct an ablation study comparing it with MonST3R on an RTX 4090 GPU. Specifically, we evaluate the memory consumption during the optimization process across varying numbers of input images. As illustrated in Figure 6, MonST3R processes all images simultaneously, leading to excessive memory usage and eventual exhaustion. In contrast, RoST3R maintains a consistent memory footprint regardless of the number of input images, highlighting its superior scalability and efficiency.

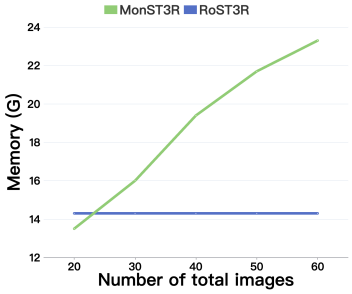

Fig. 6: **Memory consumption of different methods.**

V. CONCLUSIONS

We present RoST3R, a novel framework for reconstructing dynamic 3D scenes from streaming RGB inputs while simultaneously registering the robot into the reconstructed environment. This enables 3D-aware policy learning from purely 2D inputs. Our approach eliminates the need for depth sensors and achieves metric-accurate scene scaling, embedding both the robot and the environment within a unified coordinate frame. As a result, RoST3R significantly improves policy generalization to changes in viewpoint and camera motion, outperforming 2D methods by a wide margin. Despite its broad applicability, we evaluate RoST3R with a single policy and leave validation across diverse policies to future work.

REFERENCES

[1] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *RSS*, 2023.

[2] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[3] Z. Teed and J. Deng, "DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras," *Advances in neural information processing systems*, 2021.

[4] Z. Li, R. Tucker, F. Cole, Q. Wang, L. Jin, V. Ye, A. Kanazawa, A. Holynski, and N. Snavely, "MegaSaM: Accurate, fast and robust structure and motion from casual dynamic videos," *arXiv*, 2024.

[5] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *CVPR*, 2016.

[6] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM siggraph 2006 papers*, 2006, pp. 835–846.

[7] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.

[8] C. Tang and P. Tan, "Ba-net: Dense bundle adjustment network," *arXiv preprint arXiv:1806.04807*, 2018.

[9] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[10] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[11] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: A versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[12] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[13] J. Kopf, X. Rong, and J.-B. Huang, "Robust consistent video depth estimation," in *CVPR*, 2021.

[14] Z. Zhang, F. Cole, Z. Li, M. Rubinstein, N. Snavely, and W. T. Freeman, "Structure and motion from casual videos," in *European Conference on Computer Vision*. Springer, 2022, pp. 20–37.

[15] J. Zhang, C. Herrmann, J. Hur, V. Jampani, T. Darrell, F. Cole, D. Sun, and M.-H. Yang, "Monst3r: A simple approach for estimating geometry in the presence of motion," *arXiv*, 2024.

[16] J. Lu, T. Huang, P. Li, Z. Dou, C. Lin, Z. Cui, Z. Dong, S.-K. Yeung, W. Wang, and Y. Liu, "Align3r: Aligned monocular depth estimation for dynamic videos," *arXiv preprint arXiv:2412.03079*, 2024.

[17] Q. Wang, Y. Zhang, A. Holynski, A. A. Efros, and A. Kanazawa, "Continuous 3d perception model with persistent state," *arXiv*, 2025.

[18] V. Leroy, Y. Cabon, and J. Revaud, "Grounding image matching in 3d with mast3r," 2024.

[19] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," in *CVPR*, 2024.

[20] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," 2023.

[21] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "Rt-1: Robotics transformer for real-world control at scale," in *arXiv preprint arXiv:2212.06817*, 2022.

[22] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine, "Octo: An open-source generalist robot policy," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[23] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," in *RSS*, 2024.

[24] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," *Arxiv*, 2024.

[25] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *CoRL*, 2022.

[26] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki, "Act3d: 3d feature field transformers for multi-task robotic manipulation," 2023.

[27] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox, "Rvt: Robotic view transformer for 3d object manipulation," *arXiv:2306.14896*, 2023.

[28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

[29] H. Geng, F. Wang, S. Wei, Y. Li, B. Wang, B. An, C. T. Cheng, H. Lou, P. Li, Y.-J. Wang, Y. Liang, D. Goetting, C. Xu, H. Chen, Y. Qian, Y. Geng, J. Mao, W. Wan, M. Zhang, J. Lyu, S. Zhao, J. Zhang, J. Zhang, C. Zhao, H. Lu, Y. Ding, R. Gong, Y. Wang, Y. Kuang, R. Wu, B. Jia, C. Sferrazza, H. Dong, S. Huang, Y. Wang, J. Malik, and P. Abbeel, "Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning," *arXiv preprint arXiv:2504.18904*, 2025.

[30] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded sam: Assembling open-world models for diverse visual tasks," 2024.

[31] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[32] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[33] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "FoundationPose: Unified 6d pose estimation and tracking of novel objects," in *CVPR*, 2024.

[34] T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox, and S. Birchfield, "Camera-to-robot pose estimation from a single image," in *ICRA*, 2020.

[35] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Single-view robot pose and joint angle estimation via render & compare," in *CVPR*, 2021.

[36] S. James, Z. Ma, D. Rovick Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, 2020.

[37] T. Mu, Z. Ling, F. Xiang, D. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su, "Maniskill: Learning-from-demonstrations benchmark for generalizable manipulation skills," *CoRR*, vol. abs/2107.14483, 2021.

[38] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 3, pp. 7327–7334, 2022.

[39] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *arXiv preprint arXiv:2306.03310*, 2023.