

Lift3D Foundation Policy: Lifting 2D Large-Scale Pretrained Models for Robust 3D Robotic Manipulation

Yueru Jia^{1,2*}, Jiaming Liu^{1*,†}, Sixiang Chen^{1*}, Chenyang Gu¹, Zhilue Wang¹, Longzan Luo¹, Lily Lee¹, Pengwei Wang², Zhongyuan Wang², Renrui Zhang^{3†}, Shanghang Zhang^{1,2} 

¹State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University; ²Beijing Academy of Artificial Intelligence (BAAI); ³CUHK

* Equal contribution, [†] Project lead,  Corresponding author

Project web page: lift3d-web.github.io

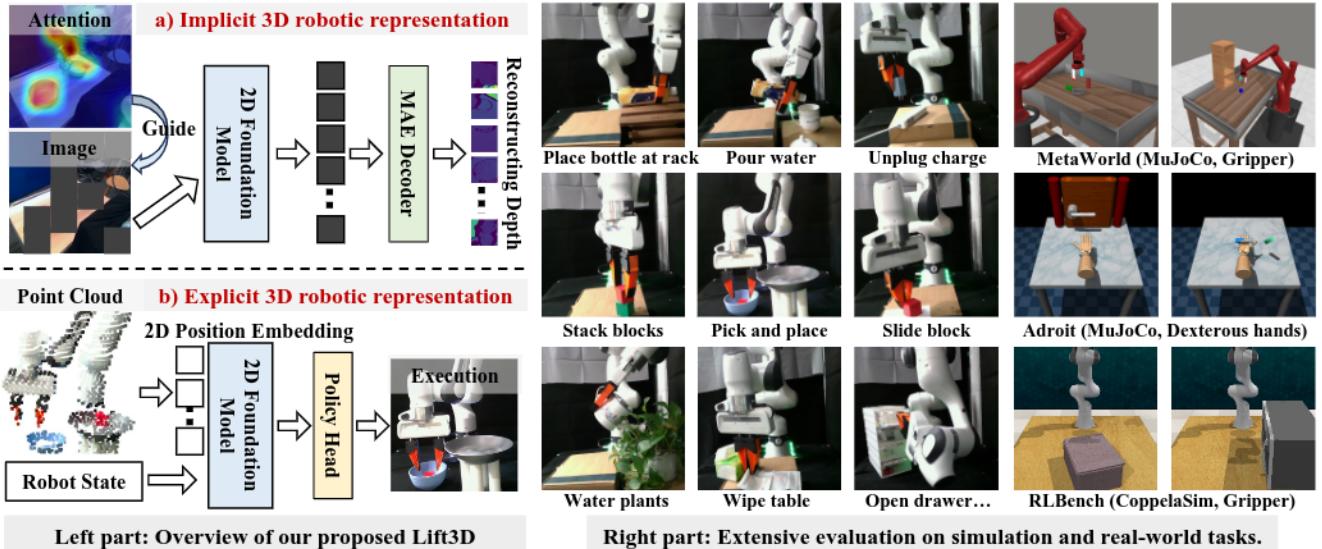


Figure 1. Lift3D empowers 2D foundation models with 3D manipulation capabilities by refining implicit 3D robotic representations through task-related affordance masking and depth reconstruction, while enhancing explicit 3D robotic representations by leveraging the pretrained 2D positional embeddings to encode point cloud. Lift3D achieves robustness and surprising effectiveness in diverse simulation and real-world tasks.

Abstract

3D geometric information is essential for manipulation tasks, as robots need to perceive the 3D environment, reason about spatial relationships, and interact with intricate spatial configurations. Recent research has increasingly focused on the explicit extraction of 3D features, while still facing challenges such as the lack of large-scale robotic 3D data and the potential loss of spatial geometry. To address these limitations, we propose the Lift3D framework, which progressively enhances 2D foundation models with implicit and explicit 3D robotic representations to construct a robust 3D manipulation policy. Specifically, we first design a task-aware masked autoencoder that masks task-relevant

affordance patches and reconstructs depth information, enhancing the 2D foundation model's implicit 3D robotic representation. After self-supervised fine-tuning, we introduce a 2D model-lifting strategy that establishes a positional mapping between the input 3D points and the positional embeddings of the 2D model. Based on the mapping, Lift3D utilizes the 2D foundation model to directly encode point cloud data, leveraging large-scale pretrained knowledge to construct explicit 3D robotic representations while minimizing spatial information loss. In experiments, Lift3D consistently outperforms previous state-of-the-art methods across several simulation benchmarks and real-world scenarios.

1. Introduction

A fundamental goal of vision-based manipulation policies is to understand the scene and predict corresponding 3D poses. Some existing methods utilize 2D images as input to directly predict 3D end-effector poses through reinforcement learning [1, 12, 21, 30, 50, 83] or imitation learning [6, 13, 18, 39, 43, 46, 94, 98]. While these approaches can effectively handle a range of manipulation tasks, they fall short of fully understanding the spatial relationships and 3D structures in the physical world [16, 19, 67, 69, 95]. In robotic manipulation, 3D geometric information is crucial for tackling complex tasks, as robots must perceive the 3D environment, reason about geometric relationships, and interact with intricate spatial configurations.

Recent research has increasingly focused on the explicit extraction of 3D feature representations in robotic manipulation tasks, which can be categorized into two groups. **On one hand, some methods directly encode point cloud data** [8, 32, 49, 69, 73, 86, 93], either training 3D policy models from scratch or fine-tuning pretrained point cloud encoders(i.e., PointNet++ [58] and PointNext [59]). However, the limited availability of large-scale robotic 3D data and foundational models constrains their generalization capabilities. Additionally, processing 3D or voxel features incurs significant computational costs, hindering scalability and practicality in real-world applications. **On the other hand, some methods involve transforming modalities**, such as lifting pretrained 2D features into 3D space [22, 36, 67, 78] or projecting 3D point clouds into multi-view images for input into 2D pretrained models [23, 24, 76, 89]. Despite showing promising performance on several downstream manipulation tasks, these modality transformations inevitably lead to a loss of spatial information, hindering robots' ability to understand 3D spatial relationships. Building on the challenges in the aforementioned 3D policies, we raise a question: “*Can we develop a 3D policy model that integrates large-scale pretrained knowledge while incorporating complete 3D spatial data input?*”

To address this question, we propose the Lift3D framework, which elevates transformer-based 2D foundation model (e.g., DINOv2 [56] or CLIP [61]) to construct robust 3D manipulation policy step by step. The key insight of Lift3D is first to enhance the implicit 3D robotic representation, followed by the explicit encoding of point cloud data for policy imitation learning. For implicit 3D robotic representation, we design a task-aware masked autoencoder (MAE) that processes 2D images and reconstructs 3D geometric information in a self-supervised manner, as shown in Figure 1 a). Specifically, we utilize large-scale unlabeled datasets from robotic manipulation [25, 55] and leverage a multimodal model (i.e., CLIP) to extract image attention maps based on task text descriptions. These attention maps are then back-projected onto the 2D input to guide the MAE

masking strategy, focusing on task-related affordance regions. Reconstructing the depth of masked tokens enhances the 3D spatial awareness of the 2D foundation model, facilitating subsequent point cloud imitation learning.

For explicit 3D robotic representation, we propose a 2D model-lifting strategy that directly leverages a 2D foundation model to encode 3D point cloud data, as shown in Figure 1 b). Specifically, inspired by the virtual camera setting in [23, 24], we first project the point cloud data onto multiple virtual planes. However, our projection process is not aimed at constructing the input for the policy model, but rather at establishing a positional correspondence between the input 3D points and the pretrained 2D positional embeddings (PEs) of each virtual plane. Guided by this positional mapping, the 2D foundation model can use its original PEs to encode the point cloud data, enabling the model to extract 3D features based on its large-scale pretrained knowledge. Unlike previous methods, Lift3D eliminates modality transformation during imitation learning, minimizing the loss of robotic spatial information, while reducing computational cost by directly leveraging the 2D foundation model for forward propagation. Through a two-stage training process, Lift3D enhances the 2D foundation model with robust 3D robotic manipulation capabilities by systematically improving both implicit and explicit 3D robotic representations.

To comprehensively evaluate our proposed Lift3D, we conduct extensive experiments across three simulation benchmarks [33, 63, 84] and several real-world scenarios, including over 30 different gripper and dexterous hand manipulation tasks, as shown in Figure 1. We compare various baseline methods, such as robotic 2D representation methods [52, 54, 61], 3D representation methods [58, 59, 96], and 3D imitation learning policies [23, 24, 36]. Lift3D consistently outperforms other methods, even when utilizing the simplest MLP policy head and single-view point cloud, demonstrating the robustness of our model's manipulation capabilities and its understanding of robotic 3D spatial awareness. For instance, Lift3D achieves improvements in the average success rates of 18.2% and 21.3% over previous state-of-the-art 3D policy methods [86] on the Meta-World and Adroit benchmarks, respectively. We also explore the model scalability across several complex tasks, gradually increasing the parameters of the 2D foundation model. In real-world experiments, Lift3D can learn novel manipulation skills with just 30 episodes per task. To evaluate the generalization capabilities of Lift3D, we incorporate different manipulated instances, background scenes, and lighting conditions from the training set into the real-world testing process. Lift3D shows strong generalization abilities, effectively leveraging the large-scale pretrained knowledge of the 2D foundation model and comprehensive 3D robotic representations. In summary, our contributions are as follows:

- We propose Lift3D, which elevates 2D foundation models

- to construct a 3D manipulation policy by systematically improving implicit and explicit 3D robotic representations.
- For implicit 3D robotic representation, we design a task-aware MAE that masks task-related affordance regions and reconstructs the depth geometric information, enhancing the 3D spatial awareness of the 2D foundation model.
 - For explicit 3D robotic representation, we propose a 2D model-lifting strategy that leverages the pretrained PEs of a 2D foundation model to encode 3D point cloud data for manipulation imitation learning.

2. Related work

Robotic Representation Learning In recent years, substantial progress has been made in the field of pretrained visual representations, with a primary focus on self-supervised learning paradigms such as contrastive learning [10, 11, 27, 48, 56], self-distillation [5, 7], and masked autoencoders (MAE) [4, 17, 28, 47, 74, 79, 90]. Additionally, multi-modal alignment approaches [61, 88] leverage large-scale paired image-language data to learn more semantically-aware representations. Building on the above research, more studies are dedicated to enhancing 2D representations in the field of robotics. Specifically, R3M [54] leverages contrastive learning to learn a universal embodied representation from diverse human video data. Vip [51] generates dense, smooth reward functions for unseen robotic tasks. MVP [62], VC-1 [52], and Voltron [35] delve into the effectiveness of the MAE strategy in robotic pretraining. [80, 87] consider the impact of multiple frames on the current state. However, 2D representations often struggle to capture the spatial context needed for complex robotic tasks, prompting other pretraining work to explore 3D robotic representations. MV-MWM [65], 3D-MVP [60], and SPA [96] leverage multi-view MAE to learn 3D visual representation. SUGAR [9] and Point Cloud Matters [95] introduce point cloud-based 3D representations, showing that these observations often improve policy performance and generalization capabilities. DPR [77] leverages depth information as auxiliary knowledge for pretraining. Unlike previous robotic pretraining methods, Lift3D fully leverages existing 2D foundation models [56, 61] by enhancing them with implicit 3D robotic representations. This approach makes the first attempt to mask task-related affordance regions and reconstruct depth information, not only improving 3D spatial awareness but also facilitating subsequent point cloud imitation learning.

Robotic Manipulation. Traditional robotic manipulation typically relies on state-based reinforcement learning [2, 20, 34, 83]. In contrast, recent approaches [13, 16, 42, 53, 75, 94] utilize visual observations as input to inform predictions. Inspired by the success of Multimodal Large Language Models [3, 26, 40, 81, 92], several Vision-Language-Action models [15, 31, 41, 43, 45, 46, 98] are designed for task planning and manipulation. However, [95] demon-

strates that explicit 3D representation is essential for complex robotic tasks, offering strong robustness to variations. Another category of works focuses on directly encoding 3D information to predict end-effector poses [32, 49, 73, 86, 93]. Anygrasp [19] employs point cloud data to learn grasp poses on a large-scale dataset. [23, 24, 76] project 3D point clouds into multi-view images for input into large-scale 2D pretrained models, while [22, 36, 67, 78] lift pretrained 2D features into 3D space. PolarNet [8] and M2T2 [85] directly utilize point clouds reconstructed from RGB-D data, processing them through an encoder-transformer combination for action prediction, while C2F-ARM [32] and PerAct [69] employ voxelized point clouds with a 3D backbone for action inference. Act3D [22] and ChainedDiffuser [78] take a different approach by representing scenes as multi-scale 3D features. 3D Diffusion Actor [37] and 3D Diffusion Policy [13] harness diffusion models to generate precise actions in 3D space. Unlike previous 3D policies, Lift3D directly encodes point cloud data using a pretrained 2D foundation model and its positional embeddings, minimizing spatial information loss and enhancing the generalization capability.

3. Lift3D Method

In Section 3.1, we introduce the problem statement of our proposed Lift3D framework. In Sections 3.2 and 3.3, we detail the technical aspects of our Task-aware MAE and 2D Model-lifting Strategy, which enhance implicit and explicit 3D robotic representations, respectively

3.1. Problem Statement

For implicit 3D robotic representation, following previous MAE methods [28, 62], we first input the masked image $I \in \mathbb{R}^{W \times H \times 3}$ into the 2D foundation model $2D_e$. The output features are then fed into a decoder $2D_d$ for depth reconstruction, $D = 2D_d(2D_e(I))$, where $D \in \mathbb{R}^{W \times H \times 1}$. This process enhances the 3D spatial awareness of the $2D_e$ model and facilitates subsequent 3D imitation learning.

For explicit 3D robotic representation, we directly utilize the 2D foundation model $2D_e$ to encode 3D point cloud data $P \in \mathbb{R}^{N \times 6}$ and the robot state R_S . We then use a simple policy head π to predict the action $a = \pi(2D_e(P, R_S))$. Following previous manipulation works [22, 23], we adopt 7-DoF action to express the end-effector pose of the robot arm, which includes 3-DoF for translation, 3-DoF for rotation, and 1-DoF for the gripper state (open or closed).

3.2. Task-aware Masked Autoencoder

Several studies [38, 52, 66] have shown that 2D foundation models exhibit strong representation and generalization capabilities across various downstream robotic tasks. Building on this, Lift3D first enhances the implicit 3D robotic representations within 2D foundation models. Existing robotic MAE reconstruction methods [52, 60, 62] employ an aggressive

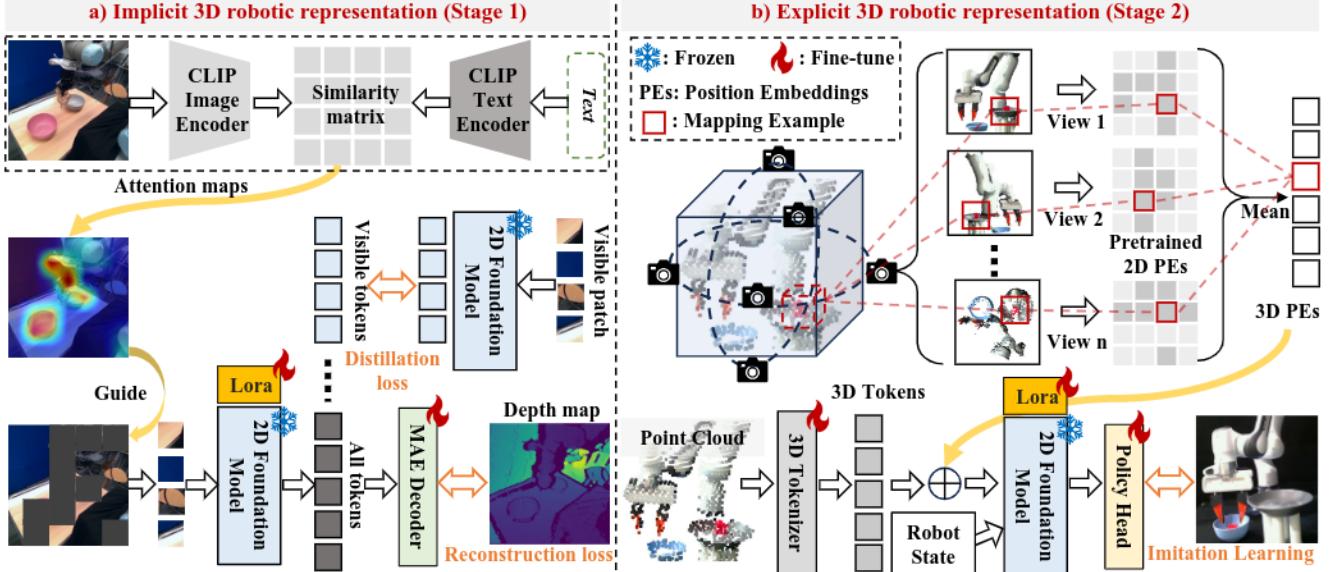


Figure 2. Overall pipeline of Lift3D. a) For implicit 3D robotic representation, we leverage CLIP [61] to offline extract image attention maps based on task descriptions, which are back-projected onto the 2D input to guide the MAE masking. We then input the visible tokens into the 2D foundation model to extract features. The masked tokens and encoded visible tokens are processed by the MAE decoder for depth reconstruction, enhancing 3D spatial awareness. Meanwhile, the encoded visible tokens are also distilled using corresponding features from the off-the-shelf pretrained model to mitigate catastrophic forgetting. b) For explicit 3D robotic representation, we first project the point cloud data onto multiple virtual planes, establishing a positional mapping between the 3D input points and the 2D positional embeddings (PEs) on each virtual plane. After mapping, we average the 2D PEs corresponding to each 3D patch to form a unified positional indicator(3D PEs), which is then integrated with the 3D tokens. These 3D tokens are generated by feeding the point cloud into a lightweight 3D tokenizer. Finally, the output features from the 2D foundation model are processed through a policy head to predict the pose for imitation learning.

masking strategy, where a large portion of the input image patches is randomly masked. However, the masked patches may mostly contain irrelevant background information, hindering the effective learning of foreground object representations. Unlike previous methods, Lift3D aims to mask task-related affordance patches and reconstruct the depth geometric information, enhancing the 3D spatial awareness of the 2D foundation model. Specifically, we leverage large-scale datasets from robotic manipulation [55] to construct our MAE training dataset, which includes 1 million training samples randomly sampled from videos, consisting of paired image and depth data. The additional details of the reconstruction dataset are shown in Appendix A. As shown in Figure 2 a), once the data is obtained, we use a multi-modal model (i.e., CLIP) to generate image attention maps based on task-specific text descriptions. For instance, the text prompt used to extract the attention map in Figure 2 is "Robot arm take the red bowl and put it in the grey bowl." These attention maps are then bilinearly resized and back-projected onto the input image to guide the MAE masking strategy. To distinguish task-related affordance tokens from background tokens, we apply a threshold (i.e., $\theta = 0.5$) to filter the attention values of all tokens. Note that the attention value for each token is computed by averaging its pixel-level values. Consistent with the masking ratio used in previous methods [28], we also randomly mask background tokens to

achieve the desired ratio (i.e., $r = 75.0\%$).

The visible (x_{vis}) tokens are fed into the 2D foundation model, then concatenated with the masked tokens (x_{mask}), which are directed to a decoder for reconstruction. The reconstruction target plays a crucial role in masked image modeling, directly influencing the learning of feature representations. Previous robotic MAE methods typically use low-level RGB information [52, 60, 62] as the reconstruction target. To enhance the 3D spatial awareness of the 2D foundation model, we reconstruct depth information (D_{target}) for task-related affordance patches and randomly selected background patches. Finally, to preserve the inherent capabilities of the foundation model, we introduce a distillation loss that constrains the distance between our model's visible token outputs and the corresponding features from the off-the-shelf pretrained model ($2D_e^{pre}$). As shown in Figure 2 a), during the stage 1 training process, we fine-tune the injected adapter [29] and decoder using reconstruction and distillation losses, which are formulated as:

$$\mathcal{L}_{implicit} = \|2D_e(x_{vis}) - 2D_e^{pre}(x_{vis})\|_1 + \quad (1)$$

$$\|2D_d(2D_e(x_{vis}) \| x_{mask}) - D_{target}\|_1 \quad (2)$$

3.3. 2D Model-lifting Strategy

After endowing the 2D foundation model with implicit 3D robotic awareness, we introduce a lifting strategy that en-

ables the 2D model to explicitly comprehend point cloud data. Recent works, whether projecting 3D point clouds into multi-view images [23, 76] or lifting 2D features into 3D space [22, 67], have faced the challenge of losing spatial information due to modality transformation. Therefore, efficiently encoding 3D data has been a key research focus in the 3D robotics field. For transformer-based 2D models, positional embeddings (PEs) play an important role, as they provide positional indicators for input tokens within the attention mechanism. However, directly creating new 3D PEs to encode 3D tokens could introduce semantic discrepancies between the pretrained 2D foundation model and the newly added 3D PEs, potentially causing a loss of large-scale pretrained knowledge.

Therefore, inspired by [23, 24, 71], we project the 3D tokens onto multiple virtual planes. Unlike previous works, our projection process is not intended to construct the model’s input. Instead, it establishes a positional correspondence between the input 3D points and the pretrained 2D PEs of each virtual plane. These 2D PEs are then used to directly encode the 3D tokens. Specifically, as shown in Figure 2 b), we transform the raw point clouds into high-dimensional space (i.e., $B \times 128 \times 768$), obtaining k (128) 3D tokens through a lightweight 3D tokenizer, following previous methods [91, 97]. The 3D tokenizer consists of farthest point sampling [57] for downsampling the number of points, the k-Nearest Neighbor algorithm for local aggregation, and learnable linear layers for feature encoding. Additionally, we use $\{C_{3D}^i\}_{i=1}^k$ to represent the 3D coordinates of each 3D token. Subsequently, each 3D coordinate are projected onto n virtual planes, obtaining corresponding 3D-to-2D coordinates $\{C_{2D}^{ij}\}_{j=1}^n$. The projection mechanism is parameter-free and efficient. We adopt a cube projection method with 6 faces, effectively capturing spatial information. The n virtual planes correspond to n original 2D PEs. Using the 3D-to-2D coordinates, we assign each 3D token to n original 2D PEs, denoted as $\{PE_{2D}(C_{2D}^{ij})\}_{j=1}^n$.

After aligning each 3D token with n 2D PEs, we simply average them to create a unified positional indicator, denoted as PE_{3D} , which is formulated as:

$$PE_{3D} = \frac{1}{n} \sum_{j=1}^n (PE_{2D}(C_{2D}^{ij})) \quad (3)$$

We incorporate PE_{3D} with the 3D tokens and input them into the 2D foundation model. In this way, we utilize the n combined original 2D PEs to encode the 3D tokens, which effectively provides diverse positional relations within 2D space and mitigates spatial information loss. The output features (i.e., $B \times 128 \times 768$) from the 2D foundation model are processed through a simple policy head to predict the pose for imitation learning. We employ a three-layer multilayer perceptron (MLP) to construct the policy head. It is important to note that our Lift3D encoder can be easily

adapted to different decoders or policy heads; the MLP head is used here for simple validation. Finally, the supervision loss is formulated as:

$$\mathcal{L}_{\text{explicit}} = \text{MSE}(T_{\text{pred}}, T_{\text{gt}}) + \left(1 - \frac{R_{\text{pred}} \cdot R_{\text{gt}}}{\|R_{\text{pred}}\| \|R_{\text{gt}}\|}\right) \quad (4)$$

$$+ \text{BCE}(G_{\text{pred}}, G_{\text{gt}}) \quad (5)$$

Where T , R , and G represent translation, rotation, and gripper state in the 7-DoF end-effector pose, respectively. As shown in Figure 2 b), during stage 2 imitation learning, we freeze the parameters of the 2D foundation model and update only the 3D tokenizer, injected adapter, and policy head. Lift3D can also operate without injecting the adapter, resulting in a minor reduction in manipulation performance. The additional validations are shown in Appendix B.4.

4. Experiments

In Sections 4.1 and 4.2, we evaluate the manipulation capability of our proposed Lift3D by presenting the experimental settings and results from both simulation and real-world tasks, respectively. The effectiveness of each component is validated through an ablation study in Section 4.3. The generalization capabilities of Lift3D are examined in Section 4.4, where we test the model on different manipulated instances, background scenes, and lighting conditions. In Section 4.5, we explore the model scalability by gradually increasing the parameters of the 2D foundation model.

4.1. Simulation Experiment

Benchmarks. We select over 30 tasks from three widely-used manipulation simulation benchmarks: MetaWorld [84] and Adroit [63] in the MuJoCo simulator, and RLBench [33] in the CoppeliaSim simulator. The point cloud data is derived from single-view RGBD data using the camera extrinsics and intrinsics. For **MetaWorld**, a tabletop environment with a Sawyer arm and two-finger gripper, we select 15 tasks of varying difficulty levels [64]. These tasks, captured from two corner camera perspectives, are categorized as follows: easy tasks include *button-press*, *drawer-open*, *reach*, *handle-pull*, *peg-unplug-side*, *lever-pull*, and *dial-turn*; medium tasks include *hammer*, *sweep-into*, *bin-picking*, *push-wall*, and *box-close*; hard and very hard tasks include *assembly*, *hand-insert*, and *shelf-place*. For **Adroit**, which focuses on dexterous hand manipulation with the same camera view as in [52], we include three tasks: *hammer*, *door*, and *pen*. For **RLBench**, which uses a Franka Panda robot and a front-view camera. Due to space constraints, the results and details of RLBench are provided in Appendix B.1.

Data collection. Scripted policies are used in MetaWorld [84], where 25 demonstrations are collected, each consisting of 200 steps. For the Adroit tasks, trajectories are obtained from agents trained with reinforcement learning

Method	Type	Input Type	MetaWorld				Mean S.R.	Adroit			Mean S.R.
			Easy (7)	Medium (5)	Hard (2)	Very Hard (1)		Hammer	Door	Pen	
CLIP [61]	2D Rep.	RGB	72.9	68.8	50.0	26.0	65.3	100	100	52	84.0
R3M [54]		RGB	84.6	66.4	83.0	36.0	75.1	100	100	56	85.3
VC-1 [52]		RGB	62.6	71.6	52.0	12.0	60.8	88	100	48	78.7
PointNet [57]	3D Rep.	PC	72.0	37.6	66.0	14.0	55.9	60	100	48	69.3
PointNet++ [58]		PC	70.3	59.6	61.0	12.0	61.6	68	100	60	76.0
PointNeXt [59]		PC	82.6	62.8	59.0	20.0	68.7	52	96	48	65.3
SPA [96]		RGB	72.6	77.4	66.0	16.0	69.5	100	100	44	81.3
DP3 [86]	3D Policy	PC	85.7	49.6	57.0	18.0	65.3	88	100	12	66.7
Lift3D (DINOV2)	Ours	PC	93.1	82.4	88.0	28.0	84.5	100	100	56	85.3
Lift3D (CLIP)		PC	94.0	78.8	82.0	42.0	83.9	100	100	64	88.0

Table 1. Comparison of manipulation success rates between Lift3D and 2D & 3D baselines in simulation benchmarks. ‘2D Rep.’ and ‘3D Rep.’ refer to robotic 2D representation and 3D representation methods, respectively. ‘PC’ indicates that the model input is point cloud.

algorithms. Specifically, DAPG [63] is applied to the *door* and *hammer* tasks, while VRL3 [72] is used for the *pen* task. We collect 100 demonstrations, each consisting of 100 steps. Demonstrations in RLBench are collected through pre-defined waypoints and the Open Motion Planning Library [70], with 100 episodes gathered, each containing several key frames.

Baselines. The innovation of Lift3D lies in systematically enhancing both implicit and explicit 3D robotic representations. To evaluate its effectiveness, we compare Lift3D with 9 methods from three categories: **1) 2D Robotic Representation Methods.** We select CLIP (ViT-base) [61], which is a 2D foundation model. Additionally, R3M [54] and VC-1 [52] are included, both of 2D robotic pretraining methods. **2) 3D Representation Methods.** Following [95], we incorporate foundational 3D models, including PointNet [57], PointNet++ [58], and PointNext [59]. Additionally, we examine SPA [96], the previous SOTA 3D robotic pretraining method. Following [52], all robotic representation methods use the same three-layer policy head and training loss as Lift3D. **3) 3D Policies.** Lift3D is compared with the previous SOTA 3D Diffusion Policy (DP3) [86] on MetaWorld and Adroit, and with RVT-2 [24] on RLBench.

Training and Evaluation Details. For fair comparisons, all baselines are trained and evaluated under the same configuration, while 3D policy methods follow their original settings. The 2D and 3D visual inputs consist of 224×224 RGB images and single-view point clouds with 1,024 points, respectively. The robot state includes end-effector pose, joint positions, and velocities, which are concatenated with the visual features. We use CLIP [61] and DINOV2 [56] (ViT-base) as our 2D foundation models. Following previous work [52], the Adam optimizer is employed with parameters $(\beta_1, \beta_2) = (0.9, 0.999)$ and a learning rate of 1e-3. A constant learning rate is used for MetaWorld, while a cosine annealing scheduler with a 0.1 warm-up factor is applied for Adroit. Each method is trained for 100 epochs, with 25 rollouts performed every 10 epochs. We report the average success rate of the best-performing policy models across training. For MetaWorld, we further average the scores across the two camera views.

Quantitative Results. In Table 1, Lift3D(CLIP) achieves an average success rate of 83.9 on the MetaWorld benchmark, with 78.8 accuracy on medium-level tasks and 82.0 accuracy on hard level tasks. Compared to other robotic representation methods, Lift3D improves the mean success rate by 8.8 over the top-performing 2D method and by 14.4 over the top-performing 3D method. In addition, compared to the previous SOTA 3D policy (DP3), Lift3D achieves an accuracy improvement of 18.6. These results demonstrate that Lift3D effectively enhances the 2D foundation model with robust manipulation capabilities, enabling a deeper understanding of robotic 3D scenes by leveraging large-scale pretrained knowledge. Furthermore, Lift3D also achieves superior performance on dexterous hand tasks compared to previous robotic representation and policy methods. Note that the DoF of the dexterous hand varies across tasks, with the *hammer*, *door*, and *pen* tasks having 26, 28, and 24 DoF, respectively. The results demonstrate that our method is also effective for more complex dexterous hand manipulation tasks, owing to the strong 3D robotic representation. Lift3D(DINOV2) also shows promising results, demonstrating the method’s practicality for other 2D foundation models. The detailed scores are shown in Appendix B.3.

4.2. Real-World Experiment

Dataset collection. In our real-world setup, we conduct experiments using a Franka Research 3 arm, with a static front view captured by an Intel RealSense L515 RGBD camera. We perform ten tasks: 1) *place bottle at rack*, 2) *pour water*, 3) *unplug charger*, 4) *stack blocks*, 5) *pick and place*, 6) *slide block*, 7) *water plants*, 8) *wipe table*, 9) *open drawer*, and 10) *close drawer*. These tasks involve various types of interacted objects and manipulation actions. For each task, 40 demonstrations are collected in diverse spatial positions, with trajectories recorded at 30 fps. We select 30 episodes and extract key frames to construct the training set for each task. Examples of input point cloud data and images are displayed in Figure 4. Additional details of the real-world dataset and assets are provided in Appendix A.

Training and Evaluation Details. The implementation details are the same as in our simulation experiments. We train

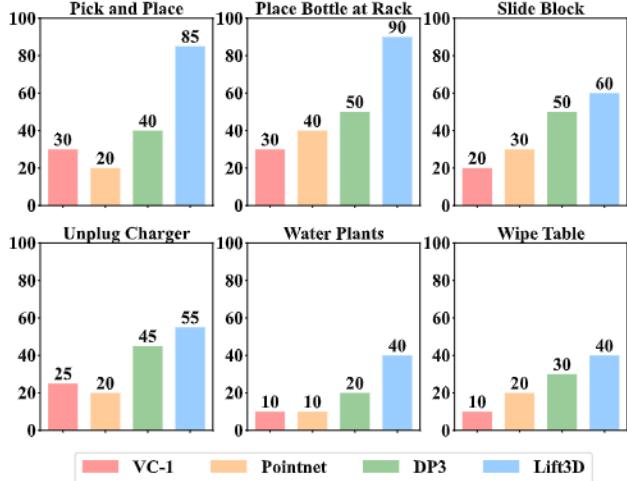


Figure 3. **Quantitative results for real robot experiments**, where the y-axis represents the manipulation success rate.

each method from scratch for each task. During the training process, point clouds and action poses in the world coordinate system are used as inputs and supervision, respectively. For evaluation, we use the model from the final epoch and evaluate it 20 times in diverse spatial positions.

Quantitative Results. As shown in Figure 3, we compare Lift3D(CLIP) with DP3 [86], VC-1 [52], and Pointnet [57]. The results demonstrate that Lift3D performs consistently well across multiple tasks. Specifically, in the *place bottle at rack* task, which requires accurate 3D position and rotation prediction, Lift3D achieves a 90 success rate. The results demonstrate that Lift3D can effectively understand 3D spatial scenes and make accurate pose predictions in the real world. Meanwhile, for complex tasks (*wipe table*), all methods face limitations in precision due to the need to manipulate deformable tissue. Despite this, Lift3D still achieves a 40 success rate. Due to space limitations, we place the other real-world experiments in Appendix B.2.

Qualitative Results. As shown in Figure 4, we visualize the manipulation process of six real-world tasks. Our method accurately predicts the continuous 7-DoF end-effector poses, allowing tasks to be completed along the trajectories. For example, in the *water plants* task, Lift3D first accurately grasps the handle of the watering can. It then smoothly lifts the can and positions it above the plants. Finally, the gripper is gradually rotated to control the flow of “water”. The demonstration video is provided in the supplementary material, and failure cases are analyzed in Appendix C.2.

4.3. Ablation Study

In Table 2, we conduct a series of ablation experiments on 2 MetaWorld simulation tasks, including *assembly* and *box-close*, and calculate the average manipulation accuracy. **For the Task-aware MAE**, in Ex2 - Ex4, we observe that Depth and RGB+Depth reconstruction outperform Ex1 with

	Task-aware MAE				2D Model-Lifting		Mean	Gain
	AMS	Depth	RGB	VD	2ML-PEs	L-PEs		
Ex1	-	-	-	-	-	-	62	+0
Ex2	-	✓	-	-	-	-	68	+6
Ex3	-	-	✓	-	-	-	63	+1
Ex4	-	✓	✓	-	-	-	67	+5
Ex5	✓	✓	-	-	-	-	72	+10
Ex6	✓	✓	-	✓	-	-	80	+18
Ex7	-	-	-	-	✓	-	86	+14
Ex8	✓	✓	-	✓	✓	-	96	+34
Ex9	✓	✓	-	✓	-	✓	90	+28

Table 2. **Ablation study.** In the Task-aware MAE, AMS refers to the affordance-guided masking strategy, Depth and RGB refer to the reconstruction targets, and VD stands for visual token distillation. In the 2D Model-Lifting, 2ML-PEs and L-PEs represent the pretrained 2D PEs constructed using our method and the newly injected learnable PEs, respectively.

success rates of 6 and 5, respectively, while RGB reconstruction alone does not show a significant improvement. This highlights the importance of reconstructing geometric information in manipulation tasks, leading us to choose Depth as our reconstruction target. By comparing Ex2 and Ex5, we find that the affordance-guided masking strategy increases the success rate by 4 compared to the random masking strategy, demonstrating that focusing on task-relevant affordance regions for learning geometric information is more efficient. Compared to Ex5, pretraining with visual token distillation (Ex6) results in an additional increase of 8, suggesting that preventing catastrophic forgetting of pretrained knowledge is essential when endowing 2D foundation models with implicit 3D robotic awareness. **For the 2D model-lifting strategy**, compared to Ex1 with image input, Ex7 introduces our lifting strategy with explicit point cloud encoding, which results in a significant improvement. The results demonstrate that 3D spatial information is crucial for achieving robust manipulation. Ex8 also shows a clear improvement over Ex7, verifying that our implicit 3D representation learning can facilitate subsequent explicit 3D imitation learning. Finally, compared to Ex8, Ex9 adopts the newly introduced PEs without pretraining and shows a 6 performance decrease, validating that our lifting strategy most effectively leverages large-scale 2D pretrained knowledge. In Appendix B.4, we explore the impact of the number and position of virtual planes, and examine the effect of parameter update manner in imitation learning.

4.4. Exploration of Generalization

Leveraging the large-scale pretrained knowledge of the 2D foundation model and comprehensive 3D robotic representations, Lift3D exhibits strong real-world generalization. As shown in Table 3, we design three real-world test scenarios, different from the training scenarios, to validate its generalization ability. **1) Different manipulated instances.** Lift3D demonstrates robustness across various manipulated objects, achieving the smallest accuracy loss. This success can primarily be attributed to the semantic understanding capabilities of the pretrained 2D foundation models. **2) Com-**

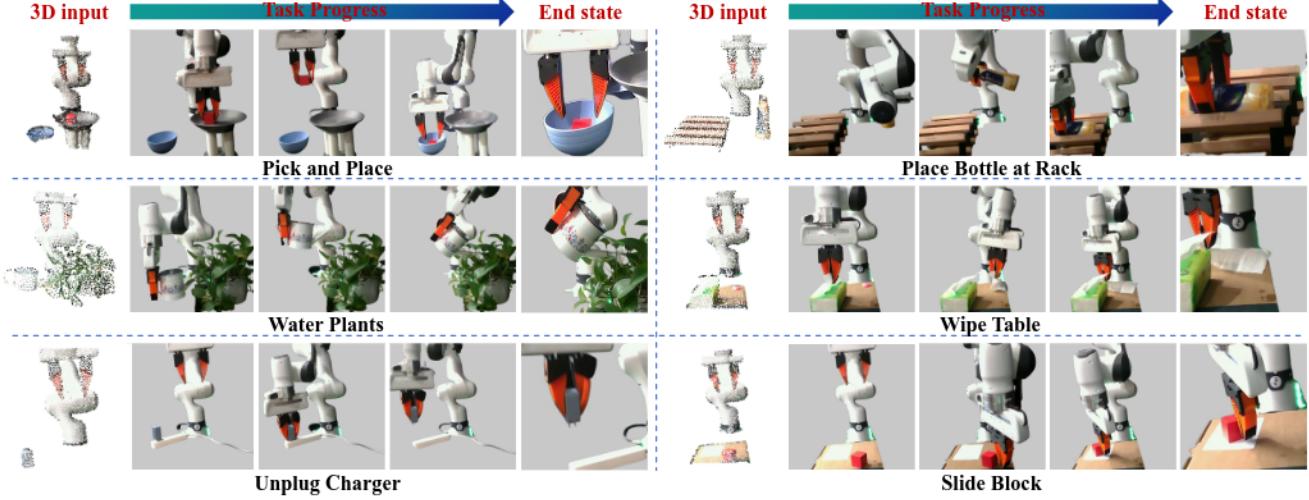
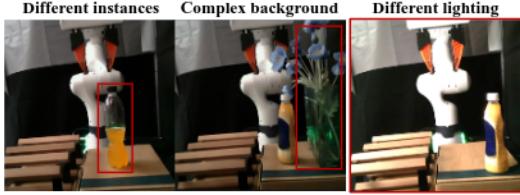


Figure 4. The qualitative results of Lift3D in real-world experiments, including the input point cloud examples, manipulation progress, and the task completion end state, are shown. More visualizations can be found in Appendix C.



Task	Pick and place			Place bottle at rack		
	Lift3D	DP3	VC-1	Lift3D	DP3	VC-1
Original Object	85	40	30	90	50	30
Background	70(-18%)	20(-50%)	20(-33%)	80(-11%)	30(-40%)	0(-100%)
Brightness	50(-41%)	20(-50%)	0(-100%)	55(-39%)	20(-60%)	0(-100%)
	60(-29%)	25 (-37%)	20(-33%)	80(-11%)	40(-20%)	20(-33%)

Table 3. **Generalization.** ‘Object’, ‘Background’, and ‘Brightness’ represent different manipulated objects, background scenes, and lighting conditions, respectively. The image above illustrates the three test scenarios, with the red box highlighting the main changes.

plex background scenes. Background distractions significantly reduce accuracy across all methods, but Lift3D shows the smallest drop, maintaining a manipulation success rate above 50. This can be attributed to the effective use of large-scale pretrained knowledge in 3D space. Additionally, the affordance-guided masking strategy enhances the model’s understanding of the spatial geometry of the foreground region through reconstruction, while minimizing the impact of background distractions. **3) Different lighting conditions.** Lighting variations affect the data distribution of 2D images and also impact depth capture, thereby influencing the point cloud data. Under the effect of lighting changes, Lift3D shows only an average accuracy drop of 20%, demonstrating its robust 3D robotic representation.

4.5. Exploration of Model Scalability

In computer vision, 2D foundation models typically enhance performance on downstream tasks as their parameters scale up [56, 61]. Building on this, we investigate whether our

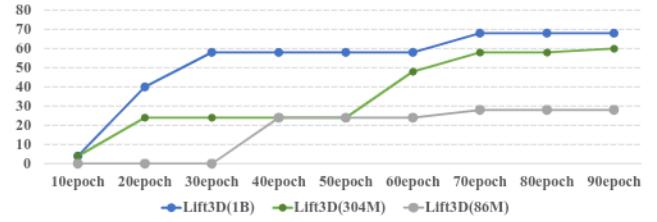


Figure 5. **Scalability.** Y-axis is the manipulation success rate.

proposed Lift3D policy also exhibits scalability. We conduct experiments on the very hard MetaWorld simulation task: *shelf-place*. For this complex task, Lift3D (DINOV2-ViT-base) achieves only 28 accuracy. The parameter count of ViT-base is only 86M, while ViT-large and ViT-giant have 304M and 1B parameters, respectively. By substituting the 2D foundation model with DINOV2-ViT-large and DINOV2-ViT-giant, Lift3D achieves 48 and 58 accuracy on the *shelf-place* task and demonstrates faster convergence, as shown in Figure 5. These improvements demonstrate the scalability of the Lift3D policy model, and the Lift3D framework is able to generate more robust manipulation policies with larger 2D foundation models.

5. Conclusion and Limitation

In this paper, we introduce Lift3D, a novel framework that integrates large-scale pretrained 2D foundation models with robust 3D manipulation capabilities. First, we design a task-aware MAE that masks task-relevant affordance regions and reconstructs depth geometric information, enhancing implicit 3D robotic representation. Second, we propose a 2D model-lifting strategy that utilizes the pretrained 2D foundation model to explicitly encode 3D point cloud data for manipulation imitation learning. Lift3D consistently outperforms existing methods in both simulation and real-world

experiments, showing strong generalization ability in diverse real-world scenarios. In terms of limitations, our Lift3D framework focuses on lifting 2D vision models to 3D manipulation tasks, which means it cannot comprehend language conditions. However, our approach can adapt to multimodal models like CLIP, enabling the integration of the Lift3D encoder with a language model and paving the way for a new 3D Vision-Language-Action model.

References

- [1] Boshi An, Yiran Geng, Kai Chen, Xiaoqi Li, Qi Dou, and Hao Dong. Rgbmanip: Monocular image-based robotic manipulation through active object pose estimation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7748–7755. IEEE, 2024. [2](#)
- [2] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. [3](#)
- [3] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023. [3](#)
- [4] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *European Conference on Computer Vision*, pages 348–367. Springer, 2022. [3](#)
- [5] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning*, pages 1298–1312. PMLR, 2022. [3](#)
- [6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. [2](#)
- [7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020. [3](#)
- [8] Shizhe Chen, Ricardo Garcia, Cordelia Schmid, and Ivan Laptev. Polarnet: 3d point clouds for language-guided robotic manipulation. *arXiv preprint arXiv:2309.15596*, 2023. [2, 3](#)
- [9] Shizhe Chen, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Sugar: Pre-training 3d visual representations for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18049–18060, 2024. [3](#)
- [10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. [3](#)
- [11] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9640–9649, 2021. [3](#)
- [12] Yuanpei Chen, Tianhao Wu, Shengjie Wang, Xidong Feng, Jiechuan Jiang, Zongqing Lu, Stephen McAleer, Hao Dong, Song-Chun Zhu, and Yaodong Yang. Towards human-level bimanual dexterous manipulation with reinforcement learning. *Advances in Neural Information Processing Systems*, 35:5150–5163, 2022. [2](#)
- [13] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023. [2, 3](#)
- [14] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024. [13](#)
- [15] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. [3](#)
- [16] Ben Eisner, Harry Zhang, and David Held. Flowbot3d: Learning 3d articulation flow to manipulate articulated objects. *arXiv preprint arXiv:2205.04382*, 2022. [2, 3](#)
- [17] David Fan, Jue Wang, Shuai Liao, Zhikang Zhang, Vimal Bhat, and Xinyu Li. Text-guided video masked autoencoder. *arXiv preprint arXiv:2408.00759*, 2024. [3](#)
- [18] Bin Fang, Shidong Jia, Di Guo, Muhua Xu, Shuhuan Wen, and Fuchun Sun. Survey of imitation learning for robotic manipulation. *International Journal of Intelligent Robotics and Applications*, 3:362–369, 2019. [2](#)
- [19] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhui Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2023. [2, 3](#)
- [20] Yiran Geng, Boshi An, Haoran Geng, Yuanpei Chen, Yaodong Yang, and Hao Dong. End-to-end affordance learning for robotic manipulation. In *International Conference on Robotics and Automation (ICRA)*, 2023. [3](#)
- [21] Yiran Geng, Boshi An, Haoran Geng, Yuanpei Chen, Yaodong Yang, and Hao Dong. Rlafford: End-to-end affordance learning for robotic manipulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5880–5886. IEEE, 2023. [2](#)
- [22] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. In *7th Annual Conference on Robot Learning*, 2023. [2, 3, 5, 14](#)
- [23] 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*, pages 694–710. PMLR, 2023. [2, 3, 5, 13](#)

- [24] Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt-2: Learning precise manipulation from few demonstrations. *arXiv preprint arXiv:2406.08545*, 2024. 2, 3, 5, 6, 13, 14
- [25] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 2
- [26] Ziyu Guo, Renrui Zhang, Xiangyang Zhu, Yiwen Tang, Xianzheng Ma, Jiaming Han, Kexin Chen, Peng Gao, Xianzhi Li, Hongsheng Li, et al. Point-bind & point-lm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following. *arXiv preprint arXiv:2309.00615*, 2023. 3
- [27] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 3
- [28] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 3, 4
- [29] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 4, 15
- [30] Hengyuan Hu, Suvir Mirchandani, and Dorsa Sadigh. Imitation bootstrapped reinforcement learning. *arXiv preprint arXiv:2311.02198*, 2023. 2
- [31] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. 3
- [32] Stephen James and Pieter Abbeel. Coarse-to-fine q-attention with learned path ranking. *arXiv preprint arXiv:2204.01571*, 2022. 2, 3
- [33] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. 2, 5, 13
- [34] Shirin Joshi, Sulabh Kumra, and Ferat Sahin. Robotic grasping using deep reinforcement learning. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1461–1466. IEEE, 2020. 3
- [35] Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023. 3
- [36] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024. 2, 3
- [37] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations, 2024. 3
- [38] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14829–14838, 2022. 3
- [39] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 2
- [40] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. 3
- [41] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023. 3
- [42] Xiaoqi Li, Yanzi Wang, Yan Shen, Ponomarenko Iaroslav, Haoran Lu, Qianxu Wang, Boshi An, Jiaming Liu, and Hao Dong. Imagemanip: Image-based robotic manipulation with affordance-guided next view selection. *arXiv preprint arXiv:2310.09069*, 2023. 3
- [43] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Maniplmm: Embodied multimodal large language model for object-centric robotic manipulation. *arXiv preprint arXiv:2312.16217*, 2023. 2, 3
- [44] Jiaming Liu, Senqiao Yang, Peidong Jia, Ming Lu, Yandong Guo, Wei Xue, and Shanghang Zhang. Vida: Homeostatic visual domain adapter for continual test time adaptation. *arXiv preprint arXiv:2306.04344*, 2023. 15
- [45] Jiaming Liu, Chenxuan Li, Guanqun Wang, Lily Lee, Kaichen Zhou, Sixiang Chen, Chuyan Xiong, Jiaxin Ge, Renrui Zhang, and Shanghang Zhang. Self-corrected multimodal large language model for end-to-end robot manipulation. *arXiv preprint arXiv:2405.17418*, 2024. 3
- [46] Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Lily Lee, Kaichen Zhou, Pengju An, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. Robomamba: Multimodal state space model for efficient robot reasoning and manipulation. *arXiv preprint arXiv:2406.04339*, 2024. 2, 3
- [47] Jiaming Liu, Ran Xu, Senqiao Yang, Renrui Zhang, Qizhe Zhang, Zehui Chen, Yandong Guo, and Shanghang Zhang. Continual-mae: Adaptive distribution masked autoencoders for continual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28653–28663, 2024. 3
- [48] Jiaming Liu, Qizhe Zhang, Xiaoqi Li, Jianing Li, Guanqun Wang, Ming Lu, Tiejun Huang, and Shanghang Zhang. Unsupervised spike depth estimation via cross-modality cross-domain knowledge transfer. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9109–9116. IEEE, 2024. 3
- [49] Minghua Liu, Xuanlin Li, Zhan Ling, Yangyan Li, and Hao Su. Frame mining: a free lunch for learning robotic manipulation from 3d point clouds. *arXiv preprint arXiv:2210.07442*, 2022. 2, 3

- [50] Andrew Lobbezoo, Yanjun Qian, and Hyock-Ju Kwon. Reinforcement learning for pick and place operations in robotics: A survey. *Robotics*, 10(3):105, 2021. 2
- [51] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022. 3
- [52] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berge, Tingfan Wu, Jay Vakil, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *Advances in Neural Information Processing Systems*, 36:655–677, 2023. 2, 3, 4, 5, 6, 7, 13, 14
- [53] Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6813–6823, 2021. 3
- [54] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022. 2, 3, 6
- [55] Abby O’Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 2, 4, 13
- [56] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 3, 6, 8
- [57] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 5, 6, 7, 13, 14, 16
- [58] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2, 6
- [59] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in neural information processing systems*, 35:23192–23204, 2022. 2, 6
- [60] Shengyi Qian, Kaichun Mo, Valts Blukis, David F Fouhey, Dieter Fox, and Ankit Goyal. 3d-mvp: 3d multiview pretraining for robotic manipulation. *arXiv preprint arXiv:2406.18158*, 2024. 3, 4
- [61] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3, 4, 6, 8, 14
- [62] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023. 3, 4
- [63] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017. 2, 5, 6
- [64] Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Conference on Robot Learning*, pages 1332–1344. PMLR, 2023. 5
- [65] Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin, and Pieter Abbeel. Multi-view masked world models for visual robotic manipulation. In *International Conference on Machine Learning*, pages 30613–30632. PMLR, 2023. 3
- [66] Jinghuan Shang, Karl Schmeckpeper, Brandon B May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. *arXiv preprint arXiv:2407.20179*, 2024. 3
- [67] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Clipor: What and where pathways for robotic manipulation. In *Conference on robot learning*, pages 894–906. PMLR, 2022. 2, 3, 5
- [68] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022. 13
- [69] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023. 2, 3
- [70] Ioan A Sucan, Mark Moll, and Lydia E Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012. 6, 13
- [71] Yiwen Tang, Ray Zhang, Jiaming Liu, Zoey Guo, Bin Zhao, Zhigang Wang, Peng Gao, Hongsheng Li, Dong Wang, and Xuelong Li. Any2point: Empowering any-modality large models for efficient 3d understanding. In *European Conference on Computer Vision*, pages 456–473. Springer, 2025. 5
- [72] Che Wang, Xufang Luo, Keith Ross, and Dongsheng Li. Vrl3: A data-driven framework for visual deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 32974–32988, 2022. 6
- [73] Chenxi Wang, Hongjie Fang, Hao-Shu Fang, and Cewu Lu. Rise: 3d perception makes real-world robot imitation simple and effective. *arXiv preprint arXiv:2404.12281*, 2024. 2, 3
- [74] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14549–14560, 2023. 3
- [75] Qianxu Wang, Haotong Zhang, Congyue Deng, Yang You, Hao Dong, Yixin Zhu, and Leonidas Guibas. Sparsedff: Sparse-view feature distillation for one-shot dexterous manipulation. *arXiv preprint arXiv:2310.16838*, 2023. 3

- [76] Weiyao Wang, Yutian Lei, Shiyu Jin, Gregory D Hager, and Liangjun Zhang. Vihe: Virtual in-hand eye transformer for 3d robotic manipulation. *arXiv preprint arXiv:2403.11461*, 2024. 2, 3, 5
- [77] Wanying Wang, Jinming Li, Yichen Zhu, Zhiyuan Xu, Zhengping Che, Yixin Peng, Chaomin Shen, Dong Liu, Feifei Feng, and Jian Tang. Visual robotic manipulation with depth-aware pretraining. *arXiv preprint arXiv:2401.09038*, 2024. 3
- [78] Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, Tsung-Wei Ke, and Katerina Fragkiadaki. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *7th Annual Conference on Robot Learning*, 2023. 2, 3
- [79] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9653–9663, 2022. 3
- [80] Jiange Yang, Bei Liu, Jianlong Fu, Bocheng Pan, Gangshan Wu, and Limin Wang. Spatiotemporal predictive pre-training for robotic motor control. *arXiv preprint arXiv:2403.05304*, 2024. 3
- [81] Senqiao Yang, Jiaming Liu, Ray Zhang, Mingjie Pan, Zoey Guo, Xiaoqi Li, Zehui Chen, Peng Gao, Yandong Guo, and Shanghang Zhang. Lidar-llm: Exploring the potential of large language models for 3d lidar understanding. *arXiv preprint arXiv:2312.14074*, 2023. 3
- [82] Senqiao Yang, Jiarui Wu, Jiaming Liu, Xiaoqi Li, Qizhe Zhang, Mingjie Pan, and Shanghang Zhang. Exploring sparse visual prompt for cross-domain semantic segmentation. *arXiv preprint arXiv:2303.09792*, 2023. 15
- [83] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021. 2, 3
- [84] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020. 2, 5
- [85] Wentao Yuan, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. M2t2: Multi-task masked transformer for object-centric pick and place. *arXiv preprint arXiv:2311.00926*, 2023. 3
- [86] 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 *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024. 2, 3, 6, 7, 14
- [87] Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang, Di Hu, Ping Luo, et al. Learning manipulation by predicting interaction. *arXiv preprint arXiv:2406.00439*, 2024. 3
- [88] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 3
- [89] Junjie Zhang, Chenjia Bai, Haoran He, Wenke Xia, Zhigang Wang, Bin Zhao, Xiu Li, and Xuelong Li. Sam-e: Leveraging visual foundation model with sequence imitation for embodied manipulation. *arXiv preprint arXiv:2405.19586*, 2024. 2
- [90] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21769–21780, 2023. 3
- [91] Renrui Zhang, Liuhui Wang, Yali Wang, Peng Gao, Hongsheng Li, and Jianbo Shi. Starting from non-parametric networks for 3d point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5344–5353, 2023. 5
- [92] Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Yichi Zhang, Ziyu Guo, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, Shanghang Zhang, et al. Mavis: Mathematical visual instruction tuning. *arXiv e-prints*, pages arXiv–2407, 2024. 3
- [93] Tong Zhang, Yingdong Hu, Jiacheng You, and Yang Gao. Leveraging locality to boost sample efficiency in robotic manipulation. *arXiv preprint arXiv:2406.10615*, 2024. 2, 3
- [94] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. 2, 3
- [95] Haoyi Zhu, Yating Wang, Di Huang, Weicai Ye, Wanli Ouyang, and Tong He. Point cloud matters: Rethinking the impact of different observation spaces on robot learning. *arXiv preprint arXiv:2402.02500*, 2024. 2, 3, 6
- [96] Haoyi Zhu, Honghui Yang, Yating Wang, Jiang Yang, Limin Wang, and Tong He. Spa: 3d spatial-awareness enables effective embodied representation. *arXiv preprint arXiv:2410.08208*, 2024. 2, 3, 6
- [97] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Jiaming Liu, Han Xiao, Chaoyou Fu, Hao Dong, and Peng Gao. No time to train: Empowering non-parametric networks for few-shot 3d scene segmentation. *arXiv preprint arXiv:2404.04050*, 2024. 5
- [98] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *7th Annual Conference on Robot Learning*, 2023. 2, 3

Due to space limitations, we provide additional details, as well as quantitative and qualitative results of our Lift3D in this supplementary material. The outline is shown below.

- **A. Additional Details (Appendix A)**
 - Details of Reconstruction Dataset
 - Additional Details of the Real-World Dataset
- **B. Additional Quantitative Experiments (Appendix B)**
 - RLBench Experiments
 - Additional Real-World Experiments
 - Detail score of MetaWorld
 - Additional Ablation Study
- **C. Additional Qualitative Experiments (Appendix C)**
 - Additional Real-World Visualization
 - Additional Failure Case Analysis

A. Additional Details

Our training dataset is divided into two parts, systematically empowering the 2D foundation model with 3D robotic manipulation capabilities. In Sections A.1, we provide additional details of the reconstruction dataset, which is used in implicit 3D robotic representations pretraining. In Sections A.2, we provide additional details of the real-world dataset, which is used in explicit 3D imitation learning.

A.1. Details of Reconstruction Dataset

Since most subsets in the open x-embodiment dataset [55] do not simultaneously contain both camera parameters and depth, we are unable to construct point cloud data for our explicit 3D imitation learning (stage 2). Therefore, we leverage this dataset to build our MAE training data. First, we select subsets that contain paired RGB, depth, and text description data. Second, we randomly sample one frame from every nine frames in each episode. As a result, the reconstruction dataset provides 1 million image-depth-text pairs. The images are used as model input, depth serves as the reconstruction target, and the text descriptions are used for task-related affordance generation. The selected subsets are:

- *tacoplay*
- *berkeley_autolab_ur5*
- *uiuc_d3field*
- *nyu_franka_play_datasetConvertedExternally_to_rlds*
- *stanford_robocookConvertedExternally_to_rlds*
- *maniskill_datasetConvertedExternally_to_rlds*

A.2. Additional Details of the Real-World Dataset

We use the Franka Research 3 (FR3) arm as the hardware platform for our real-world experiments. Due to the relatively short length of the FR3 gripper fingers, which makes it challenging to perform certain complex tasks, we 3D print and replace the original gripper with a UMI gripper [14]. We conduct ten tasks, selecting 30 episodes and extracting key frames to construct the training set for each task. The

number of key frames per task varies, as follows: **3** frames for *unplug charger*, *slide block*, *open drawer*, *close drawer*; and **4** frames for *place bottle at rack*, *pour water*, *pick and place*, *water plants*, *wipe table*. The experimental assets and environment are shown in Figure 6. During the evaluation of real-world tasks, we determine the success of each task based on human assessment. The successful states of the 10 tasks are shown in the End State images in Figure 4 of the main text and Figure 7 of the appendix.



Figure 6. **Real-world scenario.** The assets and environment configured for the real-world experiments.

B. Additional Quantitative Experiments

In Section B.1, we compare our method against other baselines using the RLbench simulator benchmark. Additional real-world experiments are presented in Section B.2, which include four real-world tasks not covered in the main text. The fine-grid success rates for each task in the MetaWorld benchmark are provided in Section B.3. Finally, Section B.4 investigates the impact of the number and positioning of virtual planes, evaluates the effect of parameter update strategies, and analyzes the influence of the 3D tokenizer’s parameter size on 3D imitation learning.

B.1. RLbench Experiments

Experiment setting. In the RLbench benchmark [33], the data are collected through pre-defined waypoints and the Open Motion Planning Library [70]. Each task consists of 100 gathered episodes. Following previous work [23, 24, 68], we use key frames to construct the training dataset. For baseline comparison, we select VC-1 [52], PointNet [57], and RVT-2 [24]. Since Lift3D and PointNet require only single-view point cloud data as input, we compare RVT-2 in two settings: using single or four different viewpoints of RGBD cameras to construct the input point cloud data.

Method	Input Type	Close box	Put rubbish in bin	Close laptop lid	Water plants	Unplug charger	Toilet seat down	Mean
VC-1 [52]	Single-view RGB	52	12	88	12	28	96	48.0
PointNet [57]	Single-view PC	52	56	88	20	36	96	58.0
RVT-2 [24]	Four-view PC	88	100	100	12	4	100	67.3
RVT-2 [24]	Single-view PC	96	100	76	16	8	96	65.3
Lift3D	Single-view PC	92	80	92	36	36	100	72.6

Table 4. **Comparison of manipulation success rates between Lift3D and 2D & 3D baselines in RLBench benchmarks.** ‘Single-view PC’ and ‘Four-view PC’ indicate the use of one or four different viewpoints of RGBD cameras to construct the input point cloud data, which does not indicate the number of virtual planes in RVT-2.

Method	Input Type	Reference	Pour Water	Stack Block	Open Drawer	Close Drawer	Mean
VC-1 [52]	Neurips 2023	RGB	35	0	10	35	20
PointNet [57]	CVPR 2017	PC	30	15	30	30	26
DP3 [86]	RSS 2024	PC	60	30	60	70	55
Lift3D	Ours	PC	85	35	60	75	63

Table 5. **Quantitative results for real robot experiments.** The training setup is consistent with the real-world experiments described in the main text. For evaluation, we use the model from the final epoch and test it 20 times across diverse spatial positions. ‘RGB’ and ‘PC’ indicate that the model input is 2D images and 3D point cloud, respectively.

Note that, many existing policies [22, 86] use single-view point cloud as input, which is a more practical and low-cost approach for real-world applications. The training details are consistent with the simulation experiments described in Section 4.1 of the main text. For a fair comparison, we ensure all methods have the same model throughput and train for the same number of iterations.

Quantitative Results. In Table 4, Lift3D(CLIP) achieves an average success rate of 72.6 on RLBench. Compared to 2D and 3D robotic representation methods, Lift3D improves the mean success rate by 24.6 and 14.6, respectively. These results demonstrate that our method effectively enhances the 3D robotic representation of the 2D foundation model. Meanwhile, when comparing Lift3D to RVT-2 under single-view point cloud input, Lift3D achieves an accuracy improvement of 7.3. Even when compared with RVT-2 using four-view point cloud input, Lift3D still achieves comparable results. These findings indicate that even with single-view point cloud data, our method demonstrates robust manipulation capabilities, highlighting the strong practicality of Lift3D. Unlike RVT-2, our Lift3D model does not utilize a language model or language prompts for task differentiation. Instead, it exclusively relies on point clouds and robot states as input. In future work, we plan to enhance our framework by incorporating a language model to better handle human language conditions, which is highly feasible and straightforward. For example, we integrate a CLIP-BERT [61] model into Lift3D(CLIP) for language text encoding.

B.2. Additional Real-World Experiments

As shown in Table 5, we present the results for the remaining four real-world tasks. **Pour Water:** This challenging task

requires complex action predictions and precise gripper rotation for controlling the bottle. Lift3D achieves a success rate of 85%, representing a 25% improvement over the previous SOTA (DP3). Lift3D successfully completes the sequence of bottle grasping, bottle moving, and precise rotation, demonstrating its significant potential in handling complex tasks. **Stack Blocks:** This task demands precise spatial understanding and position prediction. Lift3D accurately predicts the positions of both the grasping and placement blocks, achieving a 35% success rate. While the accuracy is not optimal, it still outperforms other methods, demonstrating superior 3D robotic representation. **Open/Close Drawer:** These tasks assess the model’s ability to interact with articulated objects. Lift3D achieves success rates of 60% and 75% for opening and closing drawers, respectively. It accurately predicts the grasp position and rotation for the drawer handle, as well as the precise trajectory for the opening and closing motion. The results demonstrate that Lift3D can not only predict accurate 6-DoF poses but also predict motion trajectories for articulated objects. Based on all real-world results, we evaluate the exceptional 3D robotic representation and pretraining knowledge of our Lift3D policy, which demonstrates robust manipulation capabilities across diverse real-world tasks, even with only 30 episodes of training data.

B.3. Detail Score of MetaWorld

As shown in Table 6, we present the fine-grid scores for each task in MetaWorld. The reported scores represent the average success rate across two camera views: Corner and Corner2. Lift3D (CLIP) ranks first in 8 tasks with an average success rate of 83.9%, while Lift3D (DINOv2) ranks first in 11 tasks with an average success rate of 84.5%. Notably,

Algorithm	Adroit			Mean S.R.	MetaWorld					
	Hammer	Door	Pen		Button-press	Drawer-open	Reach	Hammer	Handle-pull	Peg-unplug-side
CLIP	100	100	52	84.0	100	100	56	88	22	78
R3M	100	100	56	85.3	92	100	60	60	68	96
VC-1	88	100	48	78.7	96	100	30	88	10	50
PointNet	60	100	48	69.3	100	100	52	38	4	68
PointNet++	68	100	60	76.0	98	84	48	70	12	78
PointNeXt	52	96	48	65.3	100	100	36	50	78	92
SPA	100	100	44	81.3	100	96	56	100	36	68
DP3	88	100	12	66.7	100	100	40	100	90	98
Lift3D(Dinov2)	100	100	56	85.3	100	100	80	100	100	96
Lift3D(Clip)	100	100	64	88.0	100	100	74	94	100	98

Algorithm	MetaWorld									Mean S.R.
	Lever-pull	Dial-turn	Sweep-into	Bin-picking	Push-wall	Box-close	Assembly	Hand-insert	Shelf-place	
CLIP	72	82	40	92	64	60	64	36	26	65.3
R3M	76	100	60	60	60	92	100	66	36	75.1
VC-1	76	76	60	80	64	66	60	44	12	60.8
PointNet	86	94	24	44	36	46	100	32	14	55.9
PointNet++	94	78	42	72	28	86	96	26	12	61.6
PointNeXt	80	92	78	82	26	78	98	20	20	68.7
SPA	68	84	64	92	55	76	96	36	16	69.5
DP3	80	92	22	24	54	48	100	14	18	65.3
Lift3D(Dinov2)	76	100	80	100	40	92	100	76	28	84.5
Lift3D(Clip)	86	100	72	92	44	92	100	64	42	83.9

Table 6. Comparison of manipulation success rates between Lift3D and 2D & 3D baselines. The table presents task-specific scores for each method, covering 18 tasks in Metaworld and 3 tasks in Adroit.

Experiment	Configuration	Parameters	Mean
Virtual Planes	1 plane	-	86
	2 planes	-	92
	4 planes	-	88
	6 planes	-	96
Update Strategy	LoRA	1.01M	96
	Without LoRA	0.87M	90
	Full Fine-Tuning	116.79M	92
3D tokenizer	1 layer	0.37M	76
	2 layers	0.66M	90
	3 layers	1.01M	96
	4 layers	3.96M	94

Table 7. Ablation study results evaluating the influence of (1) the number of virtual planes for positional embeddings, (2) different parameter update strategies (LoRA, without LoRA, and full fine-tuning), and (3) the number of layers in the 3D tokenizer. Success rates highlight the advantages of the proposed configurations

Lift3D (DINOv2) achieves nearly 100% success in 7 tasks, and Lift3D (CLIP) does so in 5 tasks. These results demonstrate that Lift3D effectively enhances both the implicit and explicit 3D robotic representations of 2D foundation models, regardless of their pretraining methods. However, on the *push-wall* task, Lift3D does not achieve leading performance. By visualizing the model’s input, we find that the sparsity of the point cloud on the wall leads to inaccuracies in predicting the push position. In future work, we plan to increase the density of the input point cloud, enabling

the model to extract more precise and detailed explicit 3D robotic representations.

B.4. Additional Ablation Study

In Table 7, we present three additional ablation study on the Metaworld benchmark, which use the same task of main text (*assembly* and *box-close*), reporting the average success rate.

Number of Virtual Planes. We analyze the effect of varying the number and positions of virtual planes, which are used for positional mapping between the 3D input points and the 2D positional embeddings. The main paper reports results using six planes (top, bottom, left, right, front, and back). Here, we compare this setup with configurations using four planes (front, back, left, and right), two planes (front and back), and a single plane (front). The results, as shown in Table 7, indicate that the six-plane configuration achieves the best performance (96%), followed by two planes (92%), four planes (88%), and one plane (86%). This demonstrates that using six planes provides diverse positional relationships from multiple perspectives, better encodes the positional information of point cloud data, and minimizes the loss of spatial information.

Parameter Update Strategy for Imitation Learning. We evaluate the impact of different parameter update strategies during the imitation learning stage (Stage 2). For all strategies, we update the 3D tokenizer and policy head, both of which are randomly initialized. Our proposed method injects LoRA [29] into foundation models for parameter-efficient fine-tuning [44, 82]. Specifically, we explore two

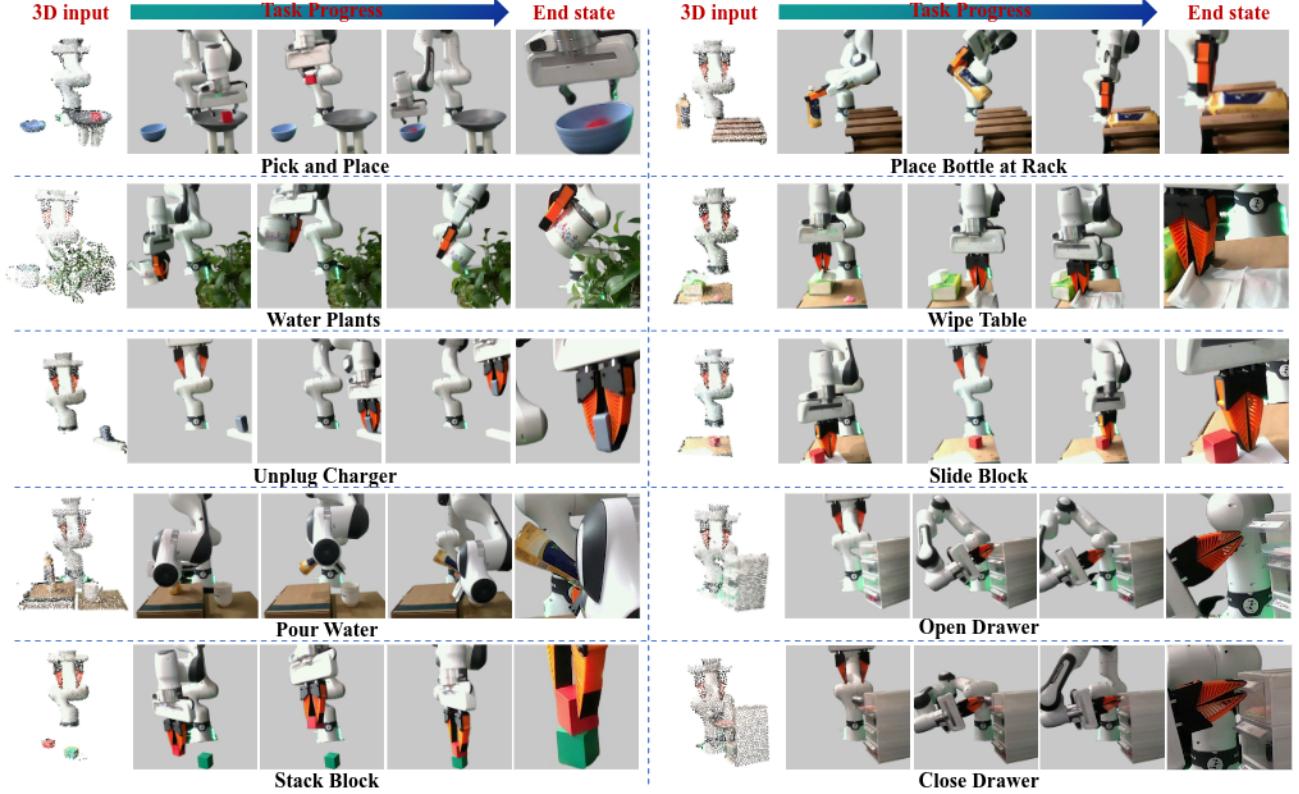


Figure 7. **The qualitative results of Lift3D in real-world experiments**, including the input point cloud examples, manipulation progress, and the task completion end state, are shown. The visualization case differs from the samples presented in the main text.

configurations: (a) full fine-tuning of all parameters and (b) excluding LoRA injections. The results, summarized in Table 7, indicate that our adopted update strategy achieves the highest mean success rate (96%), though the performance differences are minor. These findings suggest that when the policy has strong 3D robotic representations, it can deliver robust manipulation regardless of parameter update strategies. Meanwhile, Lift3D’s update strategy is highly efficient, updating only 1.01M parameters—just 1% of the total model.

Number of Layers in the 3D Tokenizer. Different layers used in the 3D tokenizer affect the module’s parameter size. Our method employs a 3-layer 3D tokenizer specifically designed for efficiently converting point clouds into 3D tokens. Each layer integrates Farthest Point Sampling [57] to reduce the number of points, the k-Nearest Neighbor algorithm ($k=64$) for local feature aggregation, and learnable linear layers for feature encoding. The main paper presents results based on the 3-layer configuration. Additionally, we conduct experiments comparing setups with 1, 2, 3, and 4 layers, where the token feature channel dimensions are set to 192, 384, 768, and 1536, respectively. For varying dimensions, we incorporate an additional linear layer to align the

channel dimensions with those required by the 2D foundation model (e.g., CLIP-ViT-base uses a channel dimension of 768). As shown in Table 7, the 3-layer configuration achieves the highest success rate (96%), outperforming the others: 4 layers (94%), 2 layers (90%), and 1 layer (76%). While both the 3-layer and 4-layer configurations demonstrate strong performance, the 3-layer setup emerges as the optimal choice, offering a better balance between accuracy and computational efficiency.

C. Additional Qualitative Experiments

In Section C.1, we visualize the manipulation process of four real-world tasks not covered in the main text. In Section C.1, we visualize the failure cases in real-world experiments and analyze the failure reasons.

C.1. Additional Real-World Visualization

As shown in Figure 7, we visualize the manipulation processes of ten real-world tasks. All visualization results are derived from our proposed Lift3D(CLIP-ViT-base) policy model. Each real-world task is specifically designed to evaluate different capabilities of the Lift3D policy model. For the first six tasks, we selected visualization cases different

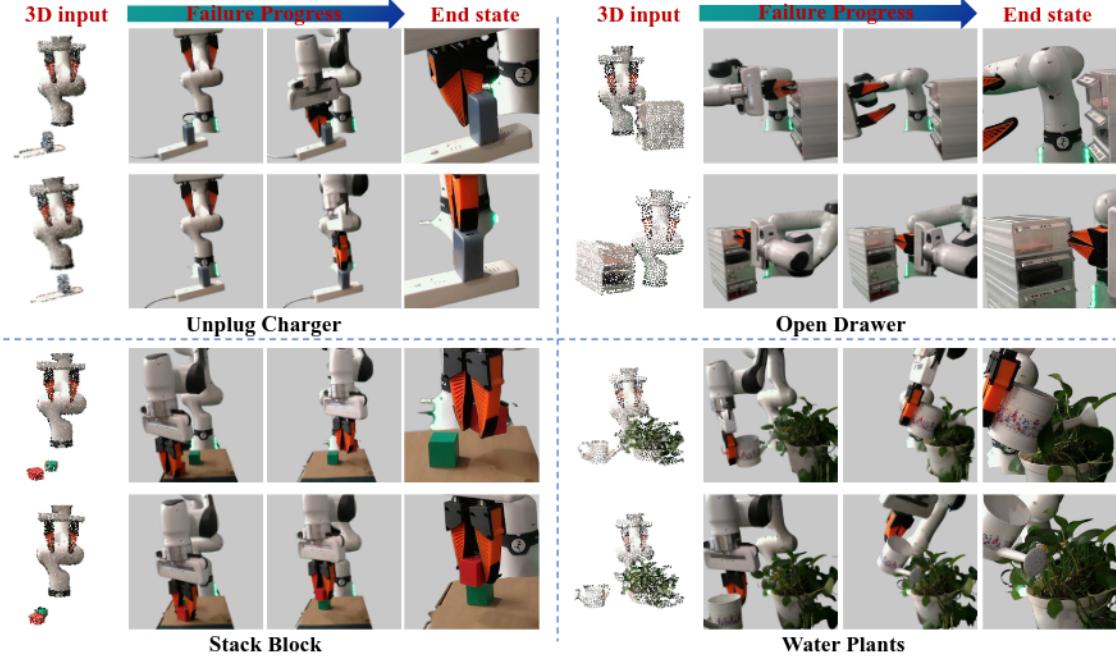


Figure 8. **The failure cases of Lift3D in real-world experiments**, including examples of input point clouds, manipulation progress, and the failure end states, are presented. The tasks include *unplug charger*, *open drawer*, *stack block*, and *water plants*.

from those presented in the main text. Our method accurately predicts 7-DoF end-effector poses, enabling tasks to be completed along the trajectories. For instance, in the *stack block* task, Lift3D first accurately grasps the red block, lifts it smoothly, and then precisely places it directly above the green block. This task highlights the model’s spatial reasoning capabilities, requiring precise perception of the red and green blocks’ positions and an accurate understanding of their spatial relationships. Demonstration videos are provided in the supplementary material.

C.2. Additional Failure Case Analysis

As shown in Figure 8, through extensive real-world experiments, we identified four primary categories of failure cases that affect the performance of Lift3D. The first category, **loss of control**, typically occurs during interactions with target objects, such as *open drawer* and *close drawer*. It is characterized by inconsistent force application when handling objects of varying weights and sudden gripper slippage on smooth surfaces. **Rotational prediction deviations** constitute the second category of failures, particularly evident in tasks requiring precise rotation control, such as *water plants*, *pour water*, and *place bottle at rack*. These failures include accumulated errors in multi-step rotational movements and incorrect rotation angles when interacting with target objects. The third category encompasses pose predictions that exceed the robot’s degree of **freedom limits**. The model occasionally predicts poses that exceed the me-

chanical limits of the Franka robotic arm, generates target positions that are unreachable due to workspace boundaries, or produces kinematically infeasible configurations during complex transitions.