

# Lift3D Policy: Lifting 2D Foundation Models for Robust 3D Robotic Manipulation

## Supplementary Material

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)**
  - RL Bench Experiments
  - Additional Real-World Experiments
  - Detail score of MetaWorld
  - Additional Ablation Study
  - Additional Scalability Experiments
- **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 [8] 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\_dataset\_converted\_externally\_to\_rlds*
- *stanford\_robotcook\_converted\_externally\_to\_rlds*
- *maniskill\_dataset\_converted\_externally\_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 [1]. 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 1. 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 2 of the appendix.



Figure 1. **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 RL Bench 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 strate-

gies, and analyzes the influence of the 3D tokenizer’s parameter size on 3D imitation learning.

### B.1. RL Bench Experiments

**Experiment setting.** In the RL Bench benchmark [6], the data are collected through pre-defined waypoints and the Open Motion Planning Library [12]. Each task consists of 100 gathered episodes. Following previous work [3, 4, 11], we use key frames to construct the training dataset. For baseline comparison, we select VC-1 [7], PointNet [9], and RVT-2 [4]. 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. Note that, many existing policies [2, 13] 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 1, Lift3D (CLIP) achieves an average success rate of 72.6 on RL Bench. 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 [10] model into Lift3D (CLIP) for language text encoding.

### B.2. Additional Real-World Experiments

As shown in Table 2, we evaluate the performance of four methods across 10 real-world tasks, covering both rigid and deformable object manipulations. The results demonstrate that **Lift3D** consistently outperforms existing approaches, highlighting its superior ability to understand 3D spatial relationships and execute precise robotic actions. Notably, Lift3D achieves a mean success rate of **62.5%**, surpassing the previous state-of-the-art DP3 by **17 percentage points**. Below, we provide a detailed analysis of four challenging

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 3, 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, 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 4, 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

Method	Input Type	Close box	Put rubbish in bin	Close laptop lid	Water plants	Unplug charger	Toilet seat down	Mean
VC-1 [7]	Single-view RGB	52	12	88	12	28	96	48.0
PointNet [9]	Single-view PC	52	56	88	20	36	96	58.0
RVT-2 [4]	Four-view PC	88	100	<b>100</b>	12	4	<b>100</b>	67.3
RVT-2 [4]	Single-view PC	<b>96</b>	<b>100</b>	76	16	8	96	65.3
<b>Lift3D</b>	Single-view PC	92	80	92	<b>36</b>	<b>36</b>	<b>100</b>	<b>72.6</b>

Table 1. **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	Pick and Place	Place Bottle	Slide Block	Unplug Charger	Water Plants	Wipe Table	Pour Water	Stack Block	Open Drawer	Close Drawer	Mean
VC-1 [7]	RGB	30	30	20	25	10	10	35	0	10	35	20.5
PointNet [9]	PC	20	40	30	20	10	20	30	15	30	30	24.5
DP3 [13]	PC	40	50	50	45	20	30	60	30	<b>60</b>	70	45.5
<b>Lift3D</b>	PC	<b>85</b>	<b>90</b>	<b>60</b>	<b>55</b>	<b>40</b>	<b>40</b>	<b>85</b>	<b>35</b>	<b>60</b>	<b>75</b>	<b>62.5</b>

Table 2. **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.

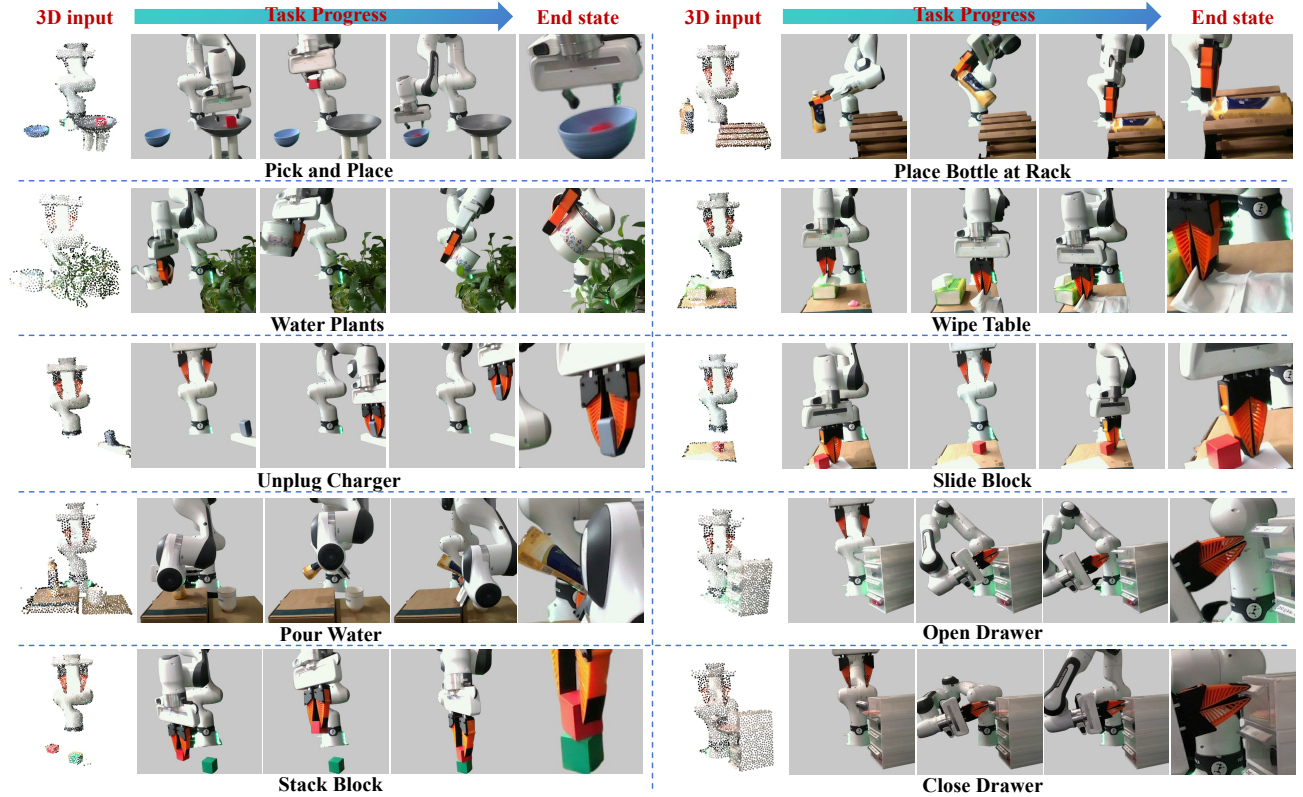


Figure 2. **The qualitative results of Lift3D in real-world experiments,** including the input point cloud examples, manipulation progress, and the task completion end state, are shown. The visualization case differs from the samples presented in the main text.

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 4, 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

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	<b>98</b>
<b>Lift3D(Dinov2)</b>	100	100	56	85.3	100	100	<b>80</b>	100	100	96
<b>Lift3D(Clip)</b>	100	100	<b>64</b>	<b>88.0</b>	100	100	74	94	100	<b>98</b>

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	<b>64</b>	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++	<b>94</b>	78	42	72	28	86	96	26	12	61.6
PointNeXt	80	92	<b>78</b>	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
<b>Lift3D(Dinov2)</b>	76	100	<b>80</b>	<b>100</b>	40	<b>92</b>	100	<b>76</b>	28	<b>84.5</b>
<b>Lift3D(Clip)</b>	86	100	72	92	44	<b>92</b>	100	64	<b>42</b>	83.9

Table 3. **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
	<b>6 planes</b>	-	<b>96</b>
Update Strategy	<b>LoRA</b>	1.01M	<b>96</b>
	Without LoRA	0.87M	90
	Full Fine-Tuning	116.79M	92
3D tokenizer	1 layer	0.37M	76
	2 layers	0.66M	90
	<b>3 layers</b>	1.01M	<b>96</b>
	4 layers	3.96M	94

Table 4. 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

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 [5] into foundation models for parameter-efficient

fine-tuning. Specifically, we explore two configurations: (a) full fine-tuning of all parameters and (b) excluding LoRA injections. The results, summarized in Table 4, 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 [9] 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 4, 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 demon-



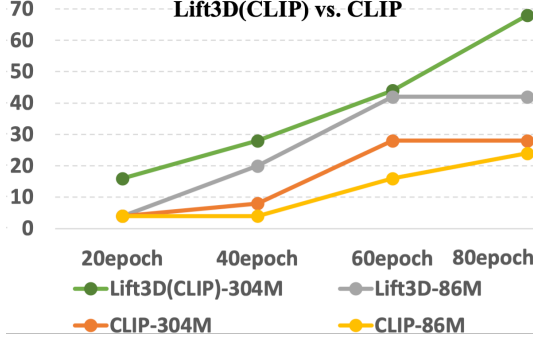


Figure 3. **Scalability.** Y-axis is the manipulation success rate.

strate strong performance, the 3-layer setup emerges as the optimal choice, offering a better balance between accuracy and computational efficiency.

**Different PE Integration Methods.** We conducted additional experiments on the assembly and close-box tasks to assess the effectiveness of different PE integration methods, including simple averaging, a learnable MLP, and max pooling. These methods achieved mean success rates of **96, 94, and 94**, respectively. Importantly, each approach preserves the alignment between 2D PEs and 3D tokens during concatenation. The results underscore the advantage of cube projection, which captures diverse positional relationships within each virtual plane, enabling a more comprehensive encoding of 3D tokens. This mitigates sensitivity to the choice of integration method, ensuring consistent and robust performance across different strategies.

### B.5. Additional Scalability Experiments

In the main text, we analyze the scalability of DINOv2 and Lift3D(DINOv2) across different model sizes, including ViT-Base, ViT-Large, and ViT-Giant. Here, we extend our study by providing additional results for CLIP and Lift3D(CLIP) using the officially provided ViT-Base and ViT-Large models, with parameter sizes of 86M and 304M, respectively.

As shown in Figure 3, Lift3D with ViT-Large achieves a score of 68, while ViT-Base reaches 42—both surpassing the original CLIP model, which only achieves 24 and 28. These results further highlight the strong scalability of Lift3D across different base models, demonstrating faster convergence and superior performance compared to their respective 2D foundation models.

## 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 2, 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 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 4, 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 mechanical limits of the Franka robotic arm, generates target positions that are unreachable due to workspace boundaries, or produces kinematically infeasible configurations during complex transitions.

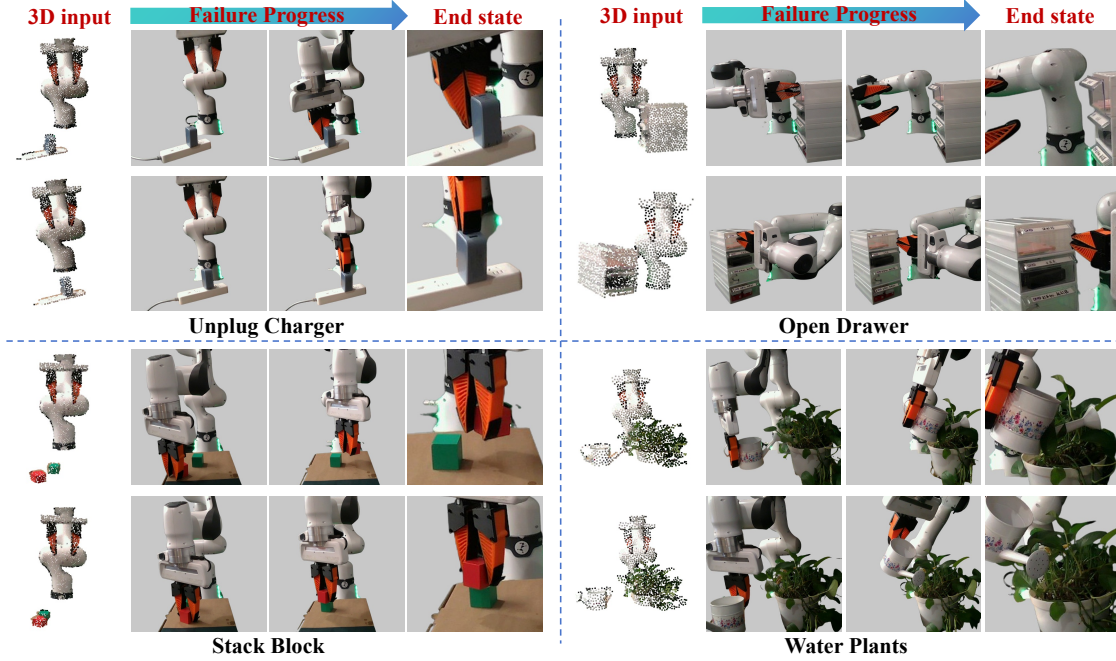


Figure 4. **The failure cases of Lift3D in real-world experiments**, including examples of input point clouds, manipulation progress, and the failure end states, are presented. The tasks include *unplug charger*, *open drawer*, *stack block*, and *water plants*.

## References

- [1] 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. 1
- [2] 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] 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
- [4] 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] 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
- [6] 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
- [7] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, 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
- [8] 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. 1
- [9] 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. 2, 3, 4
- [10] 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
- [11] 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. 2
- [12] Ioan A Sutan, Mark Moll, and Lydia E Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012. 2
- [13] 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