4D Visual Pre-training for Robot Learning

Chengkai Hou[1], **Yanjie Ze**[3], **Yankai Fu**[1], **Zeyu Gao**[4], **Songbo Hu**[2], **Yue Yu**[2]
Shanghang Zhang[1,†], **Huazhe Xu**[2,3,5,†]
[1]State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University
[2]Tsinghua University [3]Shanghai Qizhi Institute [4] CASIA [5] Shanghai AI Lab
† Corresponding author

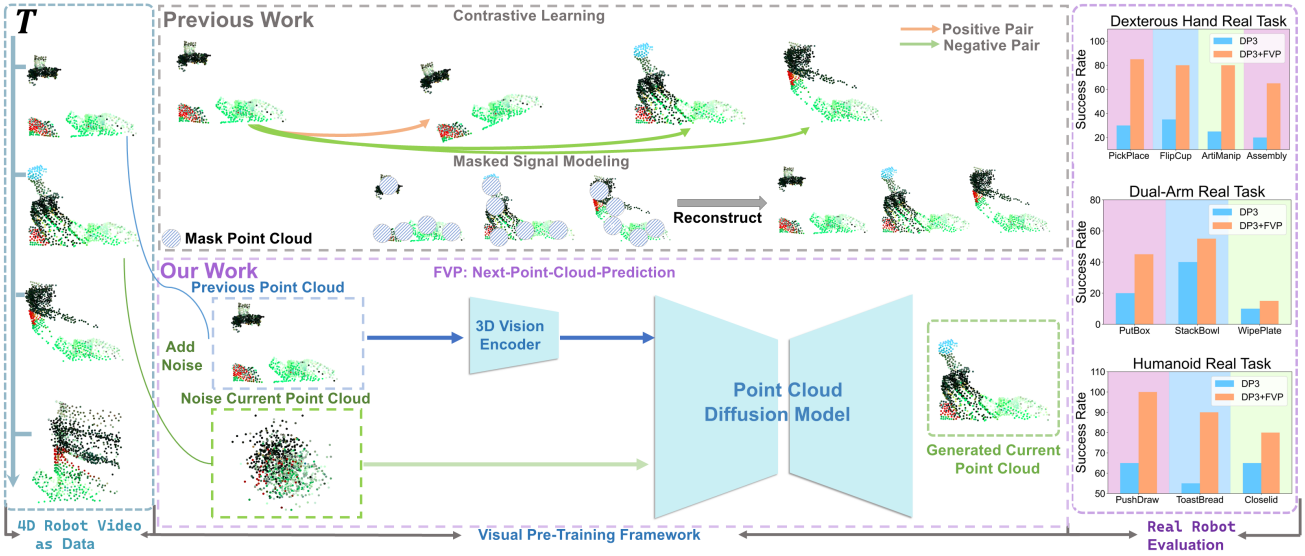

Figure 1. **FVP** is a novel 3D point cloud representation learning pipeline for robotic manipulation. Different from prior works in Contrastive Learning and Masked Signal Modeling, FVP trains 3D visual representations by leveraging the preceding frame point cloud and employing a diffusion model to predict the point cloud of the current frame.

Abstract

*General visual representations learned from web-scale datasets for robotics have achieved great success in recent years, enabling data-efficient robot learning on manipulation tasks; yet these pre-trained representations are mostly on 2D images, neglecting the inherent 3D nature of the world. However, due to the scarcity of large-scale 3D data, it is still hard to extract a universal 3D representation from web datasets. Instead, we are seeking a **general** visual pre-training framework that could improve all 3D representations as an alternative. Our framework, called FVP, is a novel **4D Visual Pre-training** framework for real-world robot learning. FVP frames the visual pre-training objective as a next-point-cloud-prediction problem, models the prediction model as a diffusion model, and pre-trains the model on the larger public datasets directly. Across twelve real-world manipulation tasks, FVP boosts the average success rate of 3D Diffusion Policy (DP3) for these tasks by **28%**. The FVP pre-trained DP3 achieves state-of-the-art performance across imitation learning methods. Moreover, the efficacy of FVP adapts across various point cloud encoders and datasets. Finally, we apply FVP to the RDT-1B, a larger Vision-Language-Action robotic model, enhancing its performance on various robot tasks. Our project page is available at: https://4d-visual-pretraining.github.io/.*

1. Introduction

Learning generalizable visual representations from large-scale datasets is crucial for robotic tasks [22, 30, 31, 49, 54]. Currently, robot representation learning is predominantly pre-trained with large-scale 2D images [19, 22, 31, 49].

However, using 3D point clouds instead of 2D images as visual sources for robotic manipulation has shown efficiency and generalization abilities on real-world robotic tasks [9, 10, 37, 44, 55, 57]. Thus, we ask: *how can we pre-train for 3D inputs and extract useful representations for robots?*

Unlike the abundance of 2D images available on the Internet, 3D point clouds are difficult to obtain from the open web. Consequently, rather than training a singular visual representation to address multiple robotic tasks, we propose a self-supervised 3D pre-training methodology that is suitable for diverse neural encoders, aimed at enhancing the performance of 3D manipulation tasks. Due to applying the diffusion model to learn the representations has yielded excellent results in visual tasks [1, 15, 46, 63], we instantiate this idea by employing a straightforward process of iteratively refining the noisy point clouds. Meanwhile, in order to acquire visual features that understand the physical environment of robots, we also incorporate the robot action information and the historical frame of robotic point cloud scene into the diffusion process.

Our method, dubbed FVP, frames the learning objective as a *next-point-cloud-prediction* problem and models the prediction network as a conditional diffusion probabilistic model. Notably, FVP directly pre-trains on the robot trajectories (*e.g.*, sequences of observation-action pairs), rendering FVP a general plug-in 4D pre-training module for all 3D imitation learning methods. FVP first embeds the history frames of the observed point cloud into the latent visual representations using a standard visual encoder such as PointNet++ [27], Point Transformer [61], and DP3 Encoder [57]. Then, conditioning on the 3D visual representations, a modified Point-Voxel Diffusion network [18, 64] gradually denoises the Gaussian noise into the point clouds of the next frame, as shown in Figure 1.

In contrast to past point cloud pre-training methods such as contrastive learning or point cloud reconstruction, FVP introduces a novel approach by predicting the next frame of point cloud. Traditional methods [13, 25, 58, 60] typically use contrastive learning where point clouds from the same time step are treated as positive pairs and those from different time steps as negative pairs; another approach is to employ point cloud reconstruction by masking portions of the point cloud (see Figure 1). However, FVP leverages the current robot observation predict the subsequent robot observation. Specifically, it enables the visual model to learn to predict the robot's next action based on the current observation. This predictive mechanism allows the visual model to better capture the motion characteristics of the robot, leading to enhanced performance in real-world robotic applications. By focusing on predicting future states, FVP enables more accurate and robust learning of dynamic behaviors—an ability that is critical for robotic tasks.

To demonstrate the effectiveness of FVP, we construct a comprehensive set of tasks comprising 12 simulation tasks and 12 real-world tasks. Simulation tasks are selected from the Adroit [32] and MetaWorld [53] benchmarks. In the real-world tasks, the robots used include single-arm robots equipped with grippers and dexterous hands, dual-arm robots, and humanoid robots. For the **Simulation** tasks, regardless of whether in-domain or out-of-domain datasets are used for pre-training, FVP-pretrained DP3 achieves the state-of-the-art performance on various simulator tasks. Specifically, it improves average task accuracy by **17%** when using in-domain datasets and by **24.7%** when using out-of-domain datasets. For the **Real** tasks, we observe that FVP could achieve **15%**$\sim$**55%** absolute improvements when built upon the state-of-the-art 3D imitation learning methods, *e.g.,* DP3 [57] and RISE [44], and largely surpass other 2D methods such as ACT [62] and Diffusion Policy [4] (see Figure 1). Moreover, we show that FVP could improve over different 3D encoders including DP3 Encoder [57], PointNet++[27], and Point Transformer [61], showing the potential in pre-training on large-scale datasets. Then, the visual models pre-trained by FVP are leveraged in the Vision-Language-Action Robotic models (VLA model), specifically RDT-1B [17]. We demonstrate through real-world tasks involving both single-arm and dual-arm robots that 3D point cloud input can effectively improve the efficiency and generalization of RDT models. Additionally, utilizing the FVP pre-trained 3D encoder on the RoboMind dataset enhances the RDT-1B model's abilities in several key areas: spatial perception, language understanding, and task generalization. We are committed to releasing the code.

2. Related Work

Visual representations for robotics. In recent years, the field of visual representations for robotics has seen significant advancements, driven by the need for robots to better understand and interact with their environments. Most works use 2D visual representations for robot control, learning from large-scale web datasets such as ImageNet [6, 36] and Ego4D [11, 22, 31, 49]. Among them, R3M [22] explores Time Contrastive Learning and Video-Language Alignment to train a universal representation for robots. MVP [49] follows the masked autoencoder paradigm and learns from Ego4D videos. VC-1 [19] scales up the model size and dataset in MVP. Recently, learning visuomotor policies from point clouds has shown great promise [37, 44, 55, 57], but a universal pre-training paradigm for robotic point cloud data remains unexplored.

Visual imitation learning provides an efficient way to teach robots human skills from human demonstrations and the learned skills could be more easily deployed in the real world compared to state-based methods [4, 37, 54, 57, 62]. Nonetheless, 2D imitation learning methods such as

ACT [62] and Diffusion Policy [4] are sensitive to camera positions and often fail to capture 3D spatial information about the objects in the environments, which highlights the necessity of 3D observations. ACT3D [9] explores the features of multi-view RGB images with a pre-trained 2D backbone and lifts them in 3D to predict the robot actions. DP3 [57] utilizes lightweight encoders to extract point cloud features, which are then fed into a diffusion model to predict the robot trajectory. Rise [44] adopts a more complex structure, including sparse convolutional networks and transformers, to encode the point cloud into point tokens and then uses these tokens to predict actions.

Diffusion models for robotics. Diffusion models are a kind of generative models that learn a denoising process by the diffusion process. They have been gaining significant popularity in the past few years due to their excellent performance in image generation [12, 34, 39, 40] and point cloud generation [21, 52, 64]. Due to the expressiveness of diffusion models, they have been applied in robotics recently, such as reinforcement learning [3, 41], imitation learning [4, 7, 23, 28, 43, 50, 57], reward learning [12, 14, 20], grasping [35, 38, 42], and motion planning [33]. Different from these works, this work provides a visual pre-training framework for robotics that is based on diffusion models.

3. Method

In this section, we describe the details of our proposed **4**D **V**isual **P**re-training (**FVP**). We begin by giving an introduction to diffusion models and then describe how FVP pre-trains 3D visual representations and applies the pre-trained representations for downstream robotic manipulation tasks.

3.1. Diffusion Models Revisited

We first give a brief introduction to the denoising diffusion probabilistic model which generates 3D point clouds through denoising process from random Gaussian noises [12, 39, 40, 64]. During training, diffusion models add a series of noises to the original point cloud X_0 as input, represented as X_T. The process of adding noise, *e.g.*, the diffusion process, is modeled as a Markov chain [16]:

$$\begin{aligned} q(X_{1:T}|X_0) &= \prod_{t=1}^{T} q(X_t|X_{t-1}), \\ q(X_t|X_{t-1}) &= \mathcal{N}(X_t; \sqrt{1-\beta_t}X_{t-1}, \beta_t \mathbf{I}). \end{aligned} \tag{1}$$

where T denotes the number of steps and $q(x_t|x_{t-1})$ is a Gaussian transition kernel, which gradually adds noise to the input with a variance schedule $\{\beta_t\}_{t=0}^{T}$. Thus, by progressively inferring the point cloud distribution, we can obtain:

$$q(X_t|X_0) = \sqrt{\bar{\alpha}_t}X_0 + \epsilon\sqrt{1-\bar{\alpha}_t}, \tag{2}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$, and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. In order to generate a recognizable object, we learn a parametrized reverse process, which denoises the noise distribution $q(X_T)$ into the target distribution $q(X_0)$. To achieve the reverse process, we utilize the network ϵ_θ to learn the reverse process $q(X_{t-1}|X_t)$. ϵ_θ: $\mathbb{R}^{N\times 3} \rightarrow \mathbb{R}^{N\times 3}$ is a diffusion model which assigns the points from Gaussian noise ball into the optimal location. Specially, at each step, we use the network to predict the offset of each point from current location and through each step iterates, the noisy point will arrive in the ideal position. Thus, the network is required to output the added noise ϵ at the most recent time step T to denoise. We use the L_2 loss $\mathcal{L}$ between the predicted noise and the ground truth $\epsilon \in \mathbb{R}^{N\times 3}$ to optimize the network:

$$\mathcal{L} = E_{\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[\|\epsilon - \epsilon_\theta(X_t, t)\|_2^2\right] \tag{3}$$

At the inference time, we reverse the diffusion process that denoises the point cloud with a standard 3D Gaussian distribution $X_T \sim N(\mathbf{0}, I_{3N})$ into a recognizable sample X_0 iteratively.

3.2. 4D Visual Pre-training on 3D Visual Representations

Demonstration collection. To pre-train 3D visual representations for downstream robotic manipulation tasks, we access the demonstrations $\mathbf{X} = \{x^0, x^1, \ldots, x^T\}$ collected from the real-world robotic tasks, where each trajectory contains T frames of observation-action pairs $x^t = (o^t, a^t)$. The observation o^t is the 3D point cloud at time t and the action is a^t the robot joint position at time t. Each task demonstrations are used to pre-train its own visual encoder. FVP is also applicable for out-of-domain pre-training using publicly available robot datasets such as Robomind, as long as they contain complete point cloud information for robotic manipulation.

Extracting 3D visual representations. FVP encodes the previous frame's point cloud o^{t-1} into a latent representation $\mathbf{z}$, which is to guide the diffusion model to predict the future frame point cloud o^t (Figure 1). The visual encoder could be implemented as any type of general 3D encoders, such as PointNet++ [27], Point Transformer [61], DP3 Encoder [57], and RISE Encoder [44]. The latent representation $\mathbf{z} \in \mathbb{R}^{N\times C_v}$, where N is the number of point clouds, C_v are the feature dimensions of point clouds.

Generating future point cloud. Conditioning on the latent representation $\mathbf{z}$, our point cloud diffusion model denoises the random Gaussian noise into the future point cloud. In particular, we project the latent representation $\mathbf{z}$ onto the current frame of point cloud with added noise o_T^t, T represents the number of added noisy steps. The input point cloud of the diffusion model is changed from $o_T^t \in \mathbb{R}^{N\times 3}$ to $o_{T,+}^t \in \mathbb{R}^{N\times(C_v+3)}$. ϵ_θ is now a new function: $\mathbb{R}^{N\times(C_v+3)} \rightarrow \mathbb{R}^{N\times 3}$ which predicts the noise ϵ from

the attached point cloud $o^t_{T,+} = [o^t_T, \mathbf{z}]$. Thus, the optimization of the loss function $\mathcal{L}$ for the neural network ϵ_θ is transformed as:

$$\mathcal{L} = E_{\epsilon \sim \mathcal{N}(0,\mathbf{I})} \left[\|\epsilon - \epsilon_\theta(o^t_{+,T}, T)\|^2_2 \right] \tag{4}$$

Downstream robotic tasks. After obtaining the pre-trained 3D visual representations, we apply them in downstream real-world robotic manipulation tasks. Given the collected expert demonstrations, we train 3D visuomotor policies such as RISE [44] and DP3 [57], which adopts point clouds as input from time step t and predict robot joint positions for time step $t + 1$. We directly replace the original visual representations with the pre-trained ones and fine-tune the visual representations and the policy backbone in an end-to-end manner during training.

4. Simulation Experiment

In our experiment, we aim to investigate how the pre-trained visual representations adopted by FVP can be utilized for downstream robotic simulation and real-world manipulation tasks. As the discrepancy between simulation environments and real-world scenarios diminishes, some standardized simulation benchmarks can serve as effective tools to validate the efficacy of FVP. Therefore, in this section, we evaluate the performance of FVP on simulation tasks from the "Adroit" and "Metaworld" benchmarks.

4.1. Simulation Benchmark

Adroit. Adroit [32] introduces a set of dexterous manipulation tasks that serve as a benchmark for assessing the capabilities of deep reinforcement learning in controlling a 24-degree-of-freedom hand. The tasks include object relocation, where a ball must be moved to a randomized target location; in-hand manipulation, requiring the repositioning of a pen to match a target orientation; door opening, involving the undoing of a latch and swinging the door open; and tool use, specifically hammering a nail into a board with variable nail positions.

MetaWorld. MetaWorld [53] is a comprehensive benchmark that encompasses 50 diverse simulated robotic manipulation tasks. These tasks are designed to challenge and evaluate the capabilities of meta-reinforcement learning and multi-task learning algorithms in acquiring new skills efficiently. The tasks involve a range of actions such as reaching, pushing, grasping, and placing objects, as well as more complex maneuvers like opening doors, windows, and drawers, turning dials, and inserting pegs into holes.

4.2. Evaluation Detail

The primary objective of FVP is to provide a novel pre-training method to enhance the performance of 3D imitation learning. To this end, our main baselines are several 3D/4D visual pre-training methods. Additionally, we also compare FVP with 2D pre-training visual models in terms of their enhancement of imitation learning. Meanwhile, to validate the effectiveness of FVP, we employ both in-domain and out-of-domain datasets for pre-training. The out-of-domain datasets contain all tasks within the current benchmark, which also include the tested tasks. For example, for the "Adroit", the in-domain dataset consists of datasets for each individual task ("Hammer", "Door", "Pen"), while the out-of-domain dataset comprises the sum of all tasks datasets on the "Adorit".

Following the DP3 testing pipline, we run 3 seeds for each experiment with seed number 0, 1, 2. For each seed, we evaluate 20 episodes every 200 training epochs and then compute the average of the highest 5 success rates. We report the mean and std of success rates across 3 seeds.

4.3. Experiment Results

In Figure 2, we demonstrate the performance of different baselines pre-trained on in-domain and out-of-domain datasets on DP3 [57]. We can observe that when pre-training on the in-domain dataset, FVP exhibits an average improvement in the success rate of 16.9% on the Adorit and the Metaworld benchmarks. When FVP adopts the out-of-domain datasets to pre-train the vision encoder, DP3 pre-trained by FVP demonstrates a significant improvement in task success rates on the Adorit and Metaworld benchmarks, especially in some difficult tasks (such as Hand Insert and Pick Out of Hole Hand Insert Disassemble). Thus, we can conclude that FVP demonstrates a more effective ability to improve the success rates of tasks in simulation compared to other pre-training methods, regardless of whether pre-training is conducted on small batches of in-domain datasets or large number of out-of-domain datasets. Meanwhile, we evaluate the performance of DP3 [57], pre-trained with FVP, against 2D imitation learning utilizing a pre-trained vision backbone in Figure 2. Despite being pre-trained on datasets exceeding size 300M, the performance of MVP and R3M in enhancing the success rate of tasks when applied to **D**iffusion **P**olicy is inferior to that of FVP pre-trained on in-domain/out-of-domain data in 3D imitation learning.

5. Real-world Experiment

Currently, 3D imitation learning gains widespread application in enabling various types of robots to execute real-world tasks. In this section, we systematically evaluate the extent to which FVP enhances the performance of single task imitation learning and vision-language-action large model(VLA model) in practical tasks. Specifically, we assess the effectiveness of FVP in improving task success rates and robustness across different robotic platforms, including the **UR5** single-arm robot with a robotic arm grip-

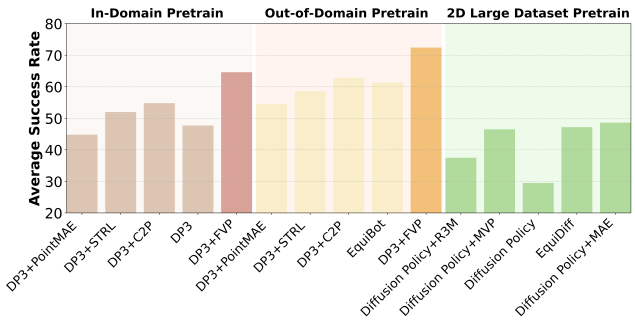

Figure 2. **Comparing FVP with more baselines in simulation.** We include various 3D pre-training methods, various 2D pre-training methods, and variants of Diffusion Policy such as EquiBot [51] and EquiDiff [45].

per and 16-Dof Leap Hand with four fingers, the **AgileX** dual-arm robot and the **TianGong** humanoid robot.

5.1. Experiment Setup

UR5 single-arm robot setup. We use the UR5 robotic arm equipped with a gripper for real-world robotic tasks. Our visual observations including images and point clouds are collected by one Intel RealSense *L515* RGB-D camera. The camera is placed in the northeast corner of the console, which is approximately 120cm by 60cm in size. For a thorough evaluation of our approach, we design two real-world tasks:

- **PickSquarel**, where the robot picks up the green square and places it in the bowl.
- **PlaceBottle**, where the robot grabs the bottle and places it on the table.

Then, we equip a UR5 single-arm with a LeapHand dexterous hand as the end effector instead of a gripper, and then we design four tasks to evaluate the effectiveness of FVP. These tasks are explained as follows:

- **PickPlace:** The dexterous hand picks up a toy chicken and places it into a blue bowl.
- **FlipCup:** The dexterous hand reaches a cup lying on the table and upright it.
- **Assembly:** The dexterous hand reaches and grasps a cylindrical cup, lifts it up and inserts it into a kettle.
- **ArtiManip:** The dexterous hand lifts the lid of a box using its thumb and gently opens it.

AgileX dual-arm robot setup. Since many operational tasks in human reality require dual-arm coordination to complete, and dual-arm coordination can achieve higher task efficiency. In our paper, we use the AgileX Cobot Magic [2] dual-arm robot setup designed based on Mobile ALOHA [8] to perform actual dual-arm tasks to validate the effectiveness of FVP. Additionally, we use the Intel RealSense *L515* RGB-D camera to record visual information during task execution. We provide a detailed description of each dual-arm manipulation task:

- **PutBox:** Both the left and right arms move the fruits from the table into the box.
- **StackBowl:** The dual arms stack two bowls on top of each other, with each arm controlling one bowl.
- **WipePlate:** The left arm holds the sponge and clean the plate picked by the right arm.

TianGong humanoid robot setup. We use the built-in cameras of TianGong humanoid robot [48] to collect visual information from real-world task scenarios, including 3D point clouds and 2D images. Simultaneously, we collect proprioceptive data, such as joint positions and actions, from the upper body of the TianGong humanoid robot. The upper body of the TianGong robot has 30 degrees of freedom (DoF), distributed across its head, arms, waist, and hands. Specifically, the head has three degrees of freedom, each arm contains seven degrees of freedom, each dexterous hand has six degrees of freedom, and the waist has one degree of freedom. To evaluate the performance of FVP in humanoid robots, we design three real-world tasks:

- **PushDraw:** The humanoid robotic arm pushes in a drawer.
- **ToastBread:** The humanoid robotic arm starts the toaster to bake bread.
- **Closelid:** The humanoid robot arm closes the garbage lid.

The visualization of the designed tasks is shown in Figure 3. Then, we introduce the data collection process for different robots. For UR5 single-arm robots with gripper, we use a keyboard interface to control the arm's movements and gripper actions. For the UR5 single-arm robot with a dexterous hand, we use HaMeR [26] to detect human hand poses with a single RealSense D435 camera. We then employ the AnyTeleop [29] framework to retarget the robot system. For the dual-arm robot, we use an auxiliary robotic arm to control the primary robotic arm to collect the dataset. For the humanoid robot, we use motion capture suits to map human movements to robot control, enabling the collection of the robot dataset. We collect **50** expert demonstrations utilized for model training. We conduct **20** trials for each experiment and report the success rate over these trials to evaluate the performance of FVP.

VLA model experiment setup. Evaluating the performance of the VLA model solely based on task success rates is not the only criterion [59]. Generalization and understanding long-range tasks are critical measures of the effectiveness of the VLA model. Figure 5 shows the four tasks we designed to investigate the spatial understanding, task transfer, language understanding, and long-horizon task performance of the VLA (Vision-Language-Action) model. These tasks include placing apples at the four corners of the space, picking up bananas and placing them on a plate, pouring water using both arms, and a long-term task that involves placing apples, pouring water, and wiping the table. Each task still requires collecting 50 demos.

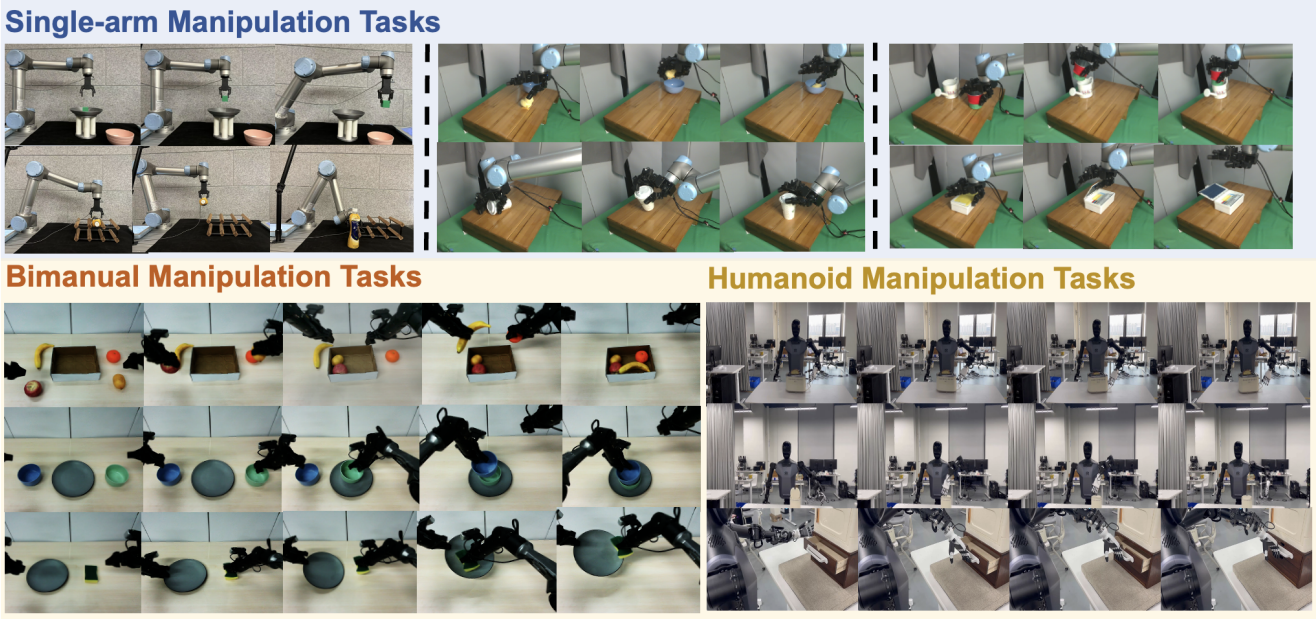

Figure 3. **Visualization of our real-world tasks.** For each task, we show several steps to understand the task process.

5.2. Q1: Can FVP-pretrained policies outperform other imitation learning methods?

We compare the DP3 and RISE pre-trained by FVP against 2D/3D imitation learning methods on our different robot tasks. Figure 4 shows that FVP pre-training approach can effectively enhance 3D imitation learning such as DP3 [57] and RISE [44]. Meanwhile, RISE pre-trained by FVP achieves the SOTA performance across these real-world tasks, largely surpassing both 2D and 3D single task imitation learning methods. Especially in the tasks of dexterous hand, FVP can notably improve the success rate of these tasks, because FVP introduces the time frames to assist visual models in understanding the complexity of motion trajectory on dexterous hand.

5.3. Q2: Can FVP outperform other pre-trained visual representations?

We select various 3D/4D pre-training methods (such as PointMAE [25], STRL [13] and C2P [60]) to train visual models for comparison with visual models pre-trained by FVP in real-world tasks. To validate the generalization of the FVP pre-training framework, we pre-train FVP and these baselines using both in-domain and out-of-domain datasets. For the out-of-domain dataset, we select the Robomind dataset [47], which contains 3D point cloud information. Figure 4 indicates that whether using an in-domain dataset or an out-of-domain dataset for pre-training, compared to PointMAE [25], STRL [13], and C2P [60], FVP pre-trained approach can learn more effective visual features, thereby aiding DP3/RISE in improving the more efficacy of real-world robotic task achievement. Vision encoders pretrained using the Robomind dataset with the FVP framework are considered as general robot vision representations. Meanwhile, we compare DP3 pre-trained by FVP with R3M [22], MVP [49] and MAE (Soup-1M+100 DoH) [5], which are the large robotic generalized models pre-trained by 2D images. We show the performance of using R3M [22], MVP [49] and MAE (Soup-1M+100 DoH) [5]-trained features in the same policy model as DP3 in Table 1. We find that FVP pre-training method is more ef-

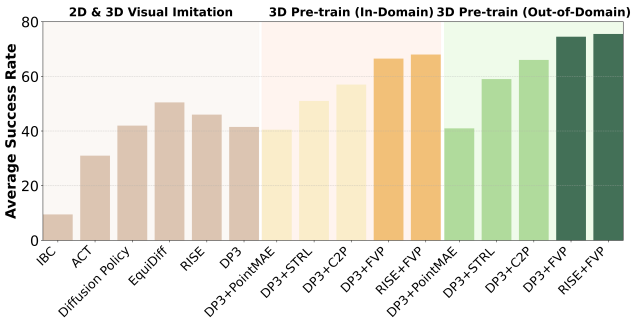

Figure 4. **Success rate (%) of imitation learning on real-world robotic tasks and 2D & 3D visual representations pre-trained by different approaches.** "DP3+FVP" and "RISE+FVP" denote the application of FVP to pretrain the visual models from DP3 and RISE, respectively. "DP3" indicates that the visual model within DP3 has not undergone pretraining. "DP3+PointMAE", "DP3+STRL", and "DP3+C2P" signify the utilization of PointMAE, STRL, and C2P to pre-train the visual model from DP3. The numbers before the comma represent the performance using in-domain datasets for pre-training, while the numbers after the comma represent the performance using out-of-domain datasets for pre-training.

fective in improving the performance of model on the real-world tasks compared to R3M [22], MVP [49] and MAE (Soup-1M+100 DoH) [5]. Similarly to the approach used in R3M [22], MVP [49] and MAE (Soup-1M+100 DoH) [5], the DP3 experiment results in the Table 1 are also pre-trained using an out-of-domain dataset. Specifically, the visual encoder from DP3 is pre-trained using the Robomind dataset [47].

Table 1. **Success rate (%) of 2D pre-trained visual representations on the diffusion policy.** We use the same policy generator as in DP3 to fine-tune R3M, MVP, and MAE (Soup-1M+100 DoH) on the six real-work tasks.

	Diffusion Policy for Robotic Action			
	R3M [22]	MVP [49]	MAE (Soup-1M+100 DoH) [5]	DP3+FVP
PickSquarel	15/20	17/20	18/20	**20/20**
PlaceBottle	13/20	15/20	15/20	**20/20**
PickPlace	14/20	16/20	16/20	**17/20**
FlipCup	14/20	**17/20**	15/20	16/20
Assembly	9/20	10/20	11/20	**13/20**
ArtiManip	11/20	14/20	14/20	**16/20**
Average	12.5/20	15.5/20	15.3/20	**16.4/20**

5.4. Q3: Can FVP improve the effectiveness of VLA models?

At present, large vision-language-action (VLA) robot models such as RDT-1B [17] rely on 2D images and robotic proprioceptive data to generate robot actions. Thus, we incorporate a point cloud encoder into the visual component of the original VLA models to support point cloud input. The point cloud visual encoder in the VLA model is the same as the one used in **iDP3** [56], featuring a pyramid-structured multi-layer fully connected network. We group tasks of the same robot type together to fine-tune RDT-1B. Table 2 shows the performance of RDT-1B, including their versions with point cloud input and pre-trained using FVP, in real-world tasks. We find that incorporating 3D point cloud input and using the FVP pre-training method significantly improves the performance of RDT-1B on real-world tasks.

Table 2. **Success rate (%) of five real-world tasks using RDT-1B with different section.** "2D Image Input" and "3D point cloud Input" refer to using only images as input and adding point clouds as additional input, respectively. "2D Image Input by R3M" and "3D encoder pretrained by FVP" refer to the experimental results using a 2D encoder pretrained with R3M and a 3D encoder pre-trained with FVP, respectively, in real-world scenarios.

Input Style	RDT-1B [27]				
	PickSquarel	PlaceBottle	PutBox	StackBowl	WipePlate
2D Image Input	12/20	10/20	6/20	8/20	3/20
2D Image Input by R3M	15/20	12/20	7/20	11/20	4/20
3D point cloud Input	14/20	12/20	9/20	13/20	4/20
3D encoder pretrained by FVP	**18/20**	**17/20**	**9/20**	**16/20**	**5/20**

5.5. Q4: Can pre-trained VLA exhibit stronger spatial understanding abilities?

We mainly examine if using 3D point cloud inputs and FVP pre-training can improve the VLA model's spatial perception capabilities. We design a pick-and-place task in which apples are placed in their designated positions based on the given instructions. We presents the visualization results of the designed tasks in Figure 5. Table 3 shows the improvement in spatial perception capabilities of the VLA model with 3D point cloud inputs and FVP pre-training.

Table 3. **Success rate (%) of RDT-1B on the different generalization tasks.** "FVP" represents FVP pre-trains the 3D encoder using the Robomind dataset.

FVP Pre-training	RDT-1B [27]		
	2D Iamge	3D PointCloud	FVP
Spatial Understanding	8/20	11/20	**14/20**
Knowledge Transfer	10/20	14/20	**16/20**
Lanugage Understanding	6/20	6/20	**7/20**
Long Horizon Task	0/20	2/20	**3/20**
Average	6/20	8.25/20	**10/20**

5.6. Q5: Can pre-trained VLA transfer their general knowledge and behavioral abilities to similar but unseen tasks?

We design a straightforward task in which the model learns to grasp a banana and place it on a plate. Subsequently, we test the model's ability to pick up an apple and place them on the plate, as depicted in the Figure 5. From Table 3, we find that due to the use of a large robotic dataset for pre-training, FVP can effectively enhance the VLA model's task transferability. Both the training and testing language inputs are *"pick up the object from the table and place it on the plate."*.

5.7. Q6: Can pre-trained VLA enhanced language understanding ability?

We aim to verify whether FVP can enhance the robustness of the VLA model in terms of language understanding. For this purpose, we design an experiment in the same scene where the task is to pour water, with language instructions to control either the left water bottle or the right water bottle to perform the pouring. Figure 5 shows the visualization results of this task. During the testing process, we input the language instructions "Pour the water from the bottle on the **Left** into the cup " and "Pour the water from the bottle on the **Right** into the cup." ten times each. Our training set further contains two types of language instructions, with an equal number of demonstrations provided for each. We find that the improvement in language understanding provided by point cloud input to the model is small (see Table 3).

5.8. Q7: Can pre-trained VLA accurately support the completion of long-horizon tasks?

We investigate whether FVP improves performance on long-range tasks. Figure 5 shows the visualization results of a long-horizon task involving multiple dual-arm operations, specifically: placing an apple on a plate, then wiping the table with a sponge, and finally pouring water into a cup. Table 3 shows that using 3D point cloud input and the

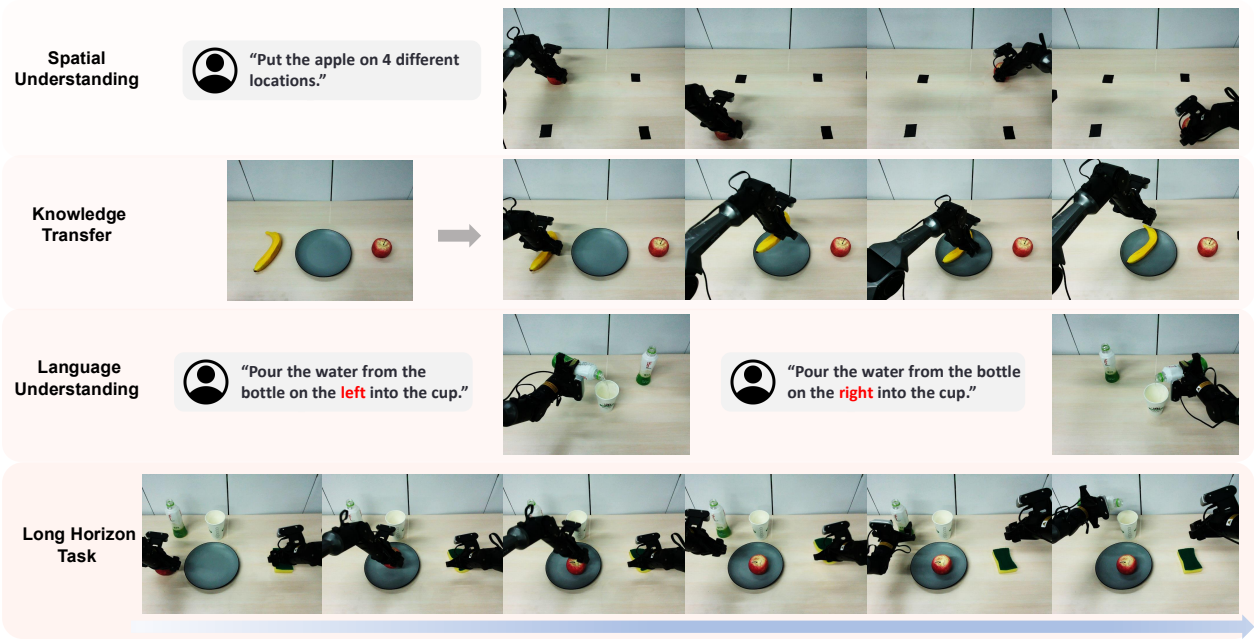

Figure 5. **Visualization of the different generalization tasks on RDT-1B.** We visualize the tasks designed to evaluate various capabilities and generalization of the RDT-1B model.

FVP pre-training method can effectively enhance the performance of the RDT-1B model on the long-horizon tasks.

Table 4. **Ablation study of DP3 pre-trained by FVP on UR5 single arm tasks.** DP3 vision encoder is pre-trained on the Robomind datasets.

	Real Tasks			
	PickSquarel	PlaceBottle	PushDraw	ToastBread
DP3+FVP	**20/20**	**20/20**	**20/20**	**16/20**
Current Frame Input	15/20	14/20	13/20	13/20
Freeze Visual Encoder	11/20	9/20	10/20	7/20

5.9. Q8: Which components of FVP are important?

To understand the contributions of each component of FVP, we conduct several ablation studies, as shown in Table 4. Specifically, we compare the **full** FVP with the **deficient** FVP, which does not history frame point cloud information. We use the current frame's point cloud instead of the historical frame point cloud to test its impact on FVP performance. Table 4 shows the success rate of DP3 pre-trained by the full/deficient FVP deployed on the several real-world robotic tasks. We can find that the information from historical frames and have a positive impact on the performance of FVP. The historical frame information plays a more significant role in the visual representations pre-trained by FVP. Table 4 shows that applying such pre-trained visual features to DP3 does not improve the model's performance. Finally, we investigate the success rate of downstream tasks when freezing the visual model during the training of DP3. Table 4 shows that freezing the visual model does not lead to an increase in the success rate of real-world tasks. We think this phenomenon is due to the gap between the out-of-domain and in-domain datasets. We also analyze the impact of using historical frames with different step sizes as the input condition on FVP's performance. Table 5 demonstrates the performance of FVP when using different historical frame point clouds as inputs in the **PickSquarel** and **PlaceBottle** task.

Table 5. Performance of **DP3+FVP** with Different Historical Frame Point Clouds in the PickSquarel and PlaceBottle Tasks

Task	1 Frame	2 Frames	3 Frames	4 Frames
PickSquarel	**20/20**	19/20	17/20	15/20
PlaceBottle	**20/20**	18/20	17/20	14/20

6. Conclusion

In this work, we introduce 4D Visual Pre-training (FVP), a visual pre-training framework for robotic manipulation, which utilizes the point cloud from history frames and robotic actions to predict the future point clouds as the learning objective, to pre-train a 3D visual representation for downstream robotic tasks. FVP is a general pre-training method for 3D imitation learning methods and we implement FVP upon DP3 and RISE, which results in state-of-the-art results across several real-world manipulation tasks. Additionally, we apply the FVP framework to the VLA (Vision-Language Action) model, which not only improve the success rate of real-world tasks but also enhance the model's generalization capabilities.

Limitations. Open-source robotics datasets, including Open-X-Embodiment [24], are available. However, these datasets lack complete camera extrinsic parameters and depth information. Thus, we do not utilize these datasets as out-of-domain data for pre-training.

7. Acknowledgment

This work was supported by the National Natural Science Foundation of China (62476011).

References

[1] Korbinian Abstreiter, Sarthak Mittal, Stefan Bauer, Bernhard Schölkopf, and Arash Mehrjou. Diffusion-based representation learning. *arXiv preprint arXiv:2105.14257*, 2021. 2

[2] AgileX Robotics. Cobot magic: An open-source robotic system. https://global.agilex.ai/products/cobot-magic, 2025. Accessed: 2025-02-22. 5

[3] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making?, 2023. 3

[4] 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

[5] Sudeep Dasari, Mohan Kumar Srirama, Unnat Jain, and Abhinav Gupta. An unbiased look at datasets for visuo-motor pre-training. In *Conference on Robot Learning*, pages 1183–1198. PMLR, 2023. 6, 7

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[7] Pete Florence, Corey Lynch, Andy Zeng, Oscar Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning, 2021. 3

[8] Z Fu, T Z Zhao, and C Finn. Mobile aloha: Learning bimanual mobile manipulation using low-cost whole-body teleoperation. In *8th Annual Conference on Robot Learning (CoRL)*, 2024. 5

[9] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: Infinite resolution action detection transformer for robotic manipulation. *arXiv preprint arXiv:2306.17817*, 2023. 2, 3

[10] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pages 694–710. PMLR, 2023. 2

[11] 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

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[13] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6535–6545, 2021. 2, 6

[14] Tao Huang, Guangqi Jiang, Yanjie Ze, and Huazhe Xu. Diffusion reward: Learning rewards via conditional video diffusion. *arXiv preprint arXiv:2312.14134*, 2023. 3

[15] Drew A Hudson, Daniel Zoran, Mateusz Malinowski, Andrew K Lampinen, Andrew Jaegle, James L McClelland, Loic Matthey, Felix Hill, and Alexander Lerchner. Soda: Bottleneck diffusion models for representation learning. *arXiv preprint arXiv:2311.17901*, 2023. 2

[16] Christopher Jarzynski. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. *Physical Review E*, 56(5):5018, 1997. 3

[17] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024. 2, 7

[18] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *Advances in neural information processing systems*, 32, 2019. 2

[19] 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, 2024. 1, 2

[20] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation, 2021. 3

[21] Benedikt Mersch, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Self-supervised point cloud prediction using 3d spatio-temporal convolutional networks. In *Conference on Robot Learning*, pages 1444–1454. PMLR, 2022. 3

[22] 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. 1, 2, 6, 7

[23] Felipe Nuti, Tim Franzmeyer, and João F. Henriques. Extracting reward functions from diffusion models, 2023. 3

[24] 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. 8

[25] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European conference on computer vision*, pages 604–621. Springer, 2022. 2, 6

[26] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9826–9836, 2024. 5

[27] 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, 3, 7

[28] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos, 2022. 3

[29] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. *arXiv preprint arXiv:2307.04577*, 2023. 5

[30] Ilija Radosavovic, Baifeng Shi, Letian Fu, Ken Goldberg, Trevor Darrell, and Jitendra Malik. Robot learning with sensorimotor pre-training. In *Conference on Robot Learning*, pages 683–693. PMLR, 2023. 1

[31] 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. 1, 2

[32] 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, 4

[33] 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, 2018. 3

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3

[35] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home, 2023. 3

[36] Rutav Shah and Vikash Kumar. Rrl: Resnet as representation for reinforcement learning. *arXiv preprint arXiv:2107.03380*, 2021. 2

[37] 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

[38] Anthony Simeonov, Ankit Goyal, Lucas Manuelli, Lin Yen-Chen, Alina Sarmiento, Alberto Rodriguez, Pulkit Agrawal, and Dieter Fox. Shelving, stacking, hanging: Relational pose diffusion for multi-modal rearrangement, 2023. 3

[39] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3

[40] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 3

[41] Kaustubh Sridhar, Souradeep Dutta, Dinesh Jayaraman, James Weimer, and Insup Lee. Memory-consistent neural networks for imitation learning, 2024. 3

[42] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. 3

[43] Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. Se(3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion, 2023. 3

[44] 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, 4, 6

[45] Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant diffusion policy. *arXiv preprint arXiv:2407.01812*, 2024. 5

[46] Chen Wei, Karttikeya Mangalam, Po-Yao Huang, Yanghao Li, Haoqi Fan, Hu Xu, Huiyu Wang, Cihang Xie, Alan Yuille, and Christoph Feichtenhofer. Diffusion models as masked autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16284–16294, 2023. 2

[47] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024. 6, 7

[48] X-Humanoid. Tiangong. `https://x-humanoid.com/bt.html`, 2025. Accessed: 2025-03-07. 5

[49] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022. 1, 2, 6, 7

[50] Ge Yan, Yueh-Hua Wu, and Xiaolong Wang. NeRFuser: Diffusion guided multi-task 3d policy learning, 2024. 3

[51] Jingyun Yang, Zi-ang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg. Equibot: Sim (3)-equivariant diffusion policy for generalizable and data efficient learning. *arXiv preprint arXiv:2407.01479*, 2024. 5

[52] Zetong Yang, Li Chen, Yanan Sun, and Hongyang Li. Visual point cloud forecasting enables scalable autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14673–14684, 2024. 3

[53] 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, 4

[54] Yanjie Ze, Nicklas Hansen, Yinbo Chen, Mohit Jain, and Xiaolong Wang. Visual reinforcement learning with self-supervised 3d representations. *IEEE Robotics and Automation Letters*, 8(5):2890–2897, 2023. 1, 2

[55] Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. In *Conference on Robot Learning*, pages 284–301. PMLR, 2023. 2

[56] Yanjie Ze, Zixuan Chen, Wenhao Wang, Tianyi Chen, Xialin He, Ying Yuan, Xue Bin Peng, and Jiajun Wu. Generalizable humanoid manipulation with improved 3d diffusion policies. *arXiv preprint arXiv:2410.10803*, 2024. 7

[57] 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 *Proceedings of Robotics: Science and Systems (RSS)*, 2024. 2, 3, 4, 6

[58] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *Advances in neural information processing systems*, 35:27061–27074, 2022. 2

[59] Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, et al. Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. *arXiv preprint arXiv:2412.18194*, 2024. 5

[60] Zhuoyang Zhang, Yuhao Dong, Yunze Liu, and Li Yi. Complete-to-partial 4d distillation for self-supervised point cloud sequence representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17661–17670, 2023. 2, 6

[61] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 2, 3

[62] 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

[63] Xiao Zheng, Xiaoshui Huang, Guofeng Mei, Yuenan Hou, Zhaoyang Lyu, Bo Dai, Wanli Ouyang, and Yongshun Gong. Point cloud pre-training with diffusion models. *arXiv preprint arXiv:2311.14960*, 2023. 2

[64] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5826–5835, 2021. 2, 3