

# PointVLA: Injecting the 3D World into Vision-Language-Action Models

Chengmeng Li<sup>1,2,\*</sup> Yichen Zhu<sup>1,\*†</sup> Junjie Wen<sup>3</sup> Yan Peng<sup>2</sup> Yixin Peng<sup>2,†</sup> Feifei Feng<sup>1</sup>

<sup>1</sup>Midea Group <sup>2</sup>Shanghai University <sup>3</sup>East China Normal University

†Corresponding author

[pointvla.github.io](https://pointvla.github.io)

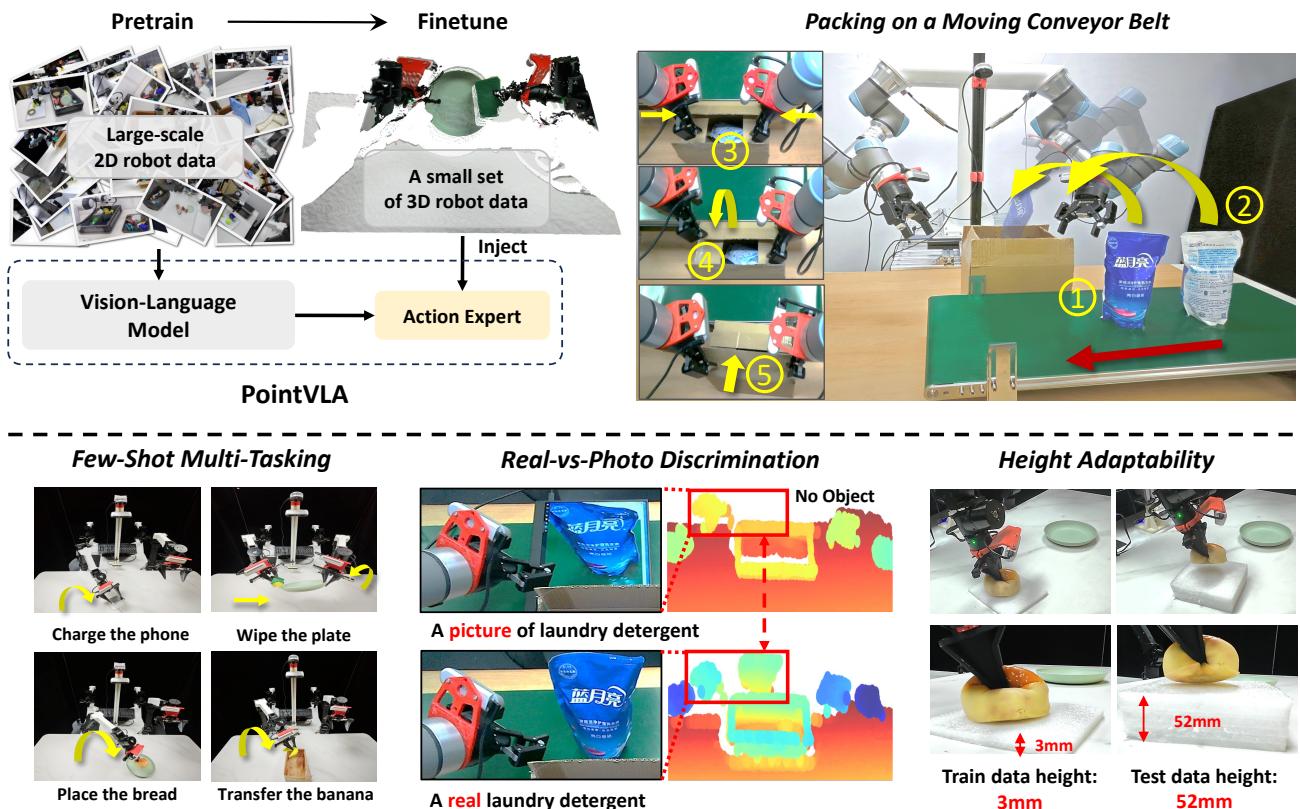


Figure 1. PointVLA builds upon the strengths of VLA, which is pre-trained on large-scale 2D robot data while incorporating 3D world into the action expert. We evaluate our method across various settings, including a challenging long-horizon packing task (top right), a few-shot multi-task setup (bottom left), and tests of the model’s real-vs-photo discrimination and height adaptability.

## Abstract

Vision-Language-Action (VLA) models excel at robotic tasks by leveraging large-scale 2D vision-language pre-training, but their reliance on RGB images limits spatial reasoning critical for real-world interaction. Retraining these models with 3D data is computationally prohibitive, while discarding existing 2D datasets wastes valuable resources. To bridge this gap, we propose PointVLA, a framework that enhances pre-trained VLAs with point cloud inputs without requiring retraining. Our method freezes

the vanilla action expert and injects 3D features via a lightweight modular block. To identify the most effective way of integrating point cloud representations, we conduct a skip-block analysis to pinpoint less useful blocks in the vanilla action expert, ensuring that 3D features are injected only into these blocks—minimizing disruption to pre-trained representations.

Extensive experiments demonstrate that PointVLA outperforms state-of-the-art 2D imitation learning methods, such as OpenVLA [25], Diffusion Policy [9] and DexVLA [46], across both simulated and real-world robotic

tasks. Specifically, we highlight several key advantages of PointVLA enabled by point cloud integration: (1) **Few-shot multi-tasking**, where PointVLA successfully performs four different tasks using only 20 demonstrations each; (2) **Real-vs-photo discrimination**, where PointVLA distinguishes real objects from their images, leveraging 3D world knowledge to improve safety and reliability; (3) **Height adaptability**, Unlike conventional 2D imitation learning methods, PointVLA enables robots to adapt to objects at varying table height that unseen in train data. Furthermore, PointVLA achieves strong performance in long-horizon tasks, such as picking and packing objects from a moving conveyor belt, showcasing its ability to generalize across complex, dynamic environments.

## 1. Introduction

Robot foundation models, particularly Vision-Language-Action (VLA) models [4, 5, 25, 45, 46], have demonstrated remarkable capabilities in enabling robots to perceive, understand, and interact with the physical world. These models leverage pre-trained vision-language models (VLMs) [3, 8, 20, 30, 42] as backbones for processing visual and linguistic information, embedding them into a shared representation space, and subsequently translating them into robot actions. This process allows robots to interact with their environment in a meaningful way. The strength of a VLA model is largely dependent on the scale and quality of its training data. For instance, OpenVLA [25] is trained on 4k hours of open-source datasets, whereas more advanced models like  $\pi_0$  utilize 10k hours of proprietary data, leading to significantly improved performance. In addition to these large-scale foundation models, numerous projects have contributed extensive datasets collected from real-world human demonstrations on physical robots. For example, AgiBot-World [6] has released a substantial dataset containing millions of trajectories demonstrating complex humanoid interactions. These pre-trained VLA models, along with open-source robotic datasets, have significantly advanced robot learning by providing a wealth of diverse and high-quality training data.

Despite these advances, most existing robot foundation models [4, 5, 21, 25, 46] are trained on 2D visual inputs [23, 35]. This represents a crucial limitation because humans perceive and interact with the world in three dimensions. The lack of comprehensive 3D spatial information in training data hinders a robot’s ability to develop a deep understanding of its environment. This is particularly critical for tasks that demand precise spatial awareness, depth perception, and object manipulation. What motivates us is the fact that many organizations have already invested heavily in foundational VLA models and large-scale 2D robot datasets. Retraining these models from scratch with 3D data

would be computationally prohibitive, and discarding valuable 2D robot data is impractical. Therefore, it is essential to explore novel frameworks that can integrate additional 3D input into pre-existing foundation robot models, a research area that has not been underexplored in the previous literature.

In this paper, we introduce PointVLA, a novel framework that integrates point clouds into pre-trained vision-language-action models. We assume that the new 3D robot data is significantly smaller than the pre-trained 2D data. In this scenario, it is critical not to disrupt the well-established 2D feature representations. To address this, we propose a 3D modular block that injects point cloud information directly into the action expert. By keeping the vision-language backbone intact, we ensure that the 2D visual-text embeddings are preserved and remain a reliable source of information. Furthermore, we aim to minimize disruptions to the action expert’s feature space. Through a skip block analysis, we identify layers that are less critical at test time. The feature embeddings of these “less useful” layers are more adaptable to the new modality. After identifying these less essential blocks, we inject the extracted 3D features via an additive approach. Our overall method maintains the integrity of the pre-trained VLA while incorporating the advantages of point cloud inputs.

We conduct extensive experiments to validate the effectiveness of our approach. For instance, on RoboTwin [32] simulation platform, our method outperforms purely 3D imitation learning methods like the 3D Diffusion Policy [51]. Additionally, we conduct real-world experiments on two types of bimanual robots: humanoid-like UR5e arms and AglieX arms resembling the Aloha platform. Furthermore, our experiments highlight several key advantages of PointVLA:

- **Few-shot multi-tasking:** PointVLA can perform four tasks following instruction, with each task trained on only 20 demonstrations. Training on such a small dataset across multiple tasks is challenging, and our approach significantly outperforms the baseline.
- **Real-vs-photo discrimination:** Real objects and their photographic representations can appear very similar in 2D images, potentially leading to confusion and safety hazards for robots. PointVLA can reliably differentiate between real objects and their images, avoiding deception by unreal objects.
- **Height adaptability:** PointVLA can adjust robot actions in response to changes in table height (e.g., grasping the same item on a much higher table), a scenario where traditional 2D VLA models typically fail.

Furthermore, we tackle challenging, long-horizon tasks such as picking multiple items from a moving conveyor belt and packing them into a box. These experiments demonstrate the robust performance and generalizability of our

proposed PointVLA framework across diverse scenarios, suggesting a promising direction for integrating additional modalities into pre-trained VLA models.

## 2. Related Works

**Vision-Language-Action models.** Recent research has increasingly focused on developing generalist robot policies trained on large-scale robotic learning datasets [11, 14, 23, 27, 35]. Vision-Language-Action (VLA) models have emerged as a promising approach for training such policies [4, 9, 12, 13, 24, 33, 36, 40, 45, 46, 48, 54, 55]. VLAs extend vision-language models (VLMs)—pre-trained on massive internet-scale image and text datasets [1, 8, 20, 28–30, 42, 53, 58, 59]—to robotic control [44]. This approach provides several key advantages: leveraging large-scale vision-language model backbones with billions of parameters enables effective learning from vast robotic datasets, while reusing pre-trained weights from internet-scale data enhances VLAs’ ability to interpret diverse language commands and generalize to novel objects and environments, making them highly adaptable for real-world robotic applications.

**Robot learning with 3D modalities.** Learning robust visuomotor policy in 3D scene [7, 15–17, 19, 22, 37, 39, 41, 49–52] is an important domain in robot learning. Existing approaches like 3DVLA [17] have proposed comprehensive frameworks that integrate diverse 3D tasks, such as generalization, visual question answering (VQA), 3D scene understanding, and robot control, into unified vision-language-action models. However, a limitation of 3DVLA is its reliance on simulation for robot control experiments, which presents a significant sim-to-real gap. Other works, such as 3D diffusion policies [51], have demonstrated that using extrinsic 3D input (e.g., from external cameras) can improve model generalization to varying lighting conditions and object attributes. iDP3 [50] further enhanced the 3D visual encoder and applied it to humanoid robots, achieving robust performance in diverse environments with both egocentric and external camera views. However, discarding existing 2D robot data or completely retraining the foundation model with added 3D visual input would be computationally expensive and resource-intensive. A more practical solution is to develop a method that integrates 3D visual input as a supplementary knowledge source into a well-pretrained foundation model, thereby reaping the benefits of new modalities without compromising the performance of the trained model.

## 3. Methodology

### 3.1. Preliminaries: Vision-Language-Action Models

Vision-Language-Action (VLA) models are driving a significant shift in real-world robot learning. Their power originates from the underlying Vision-Language Model (VLM), a robust backbone trained on vast internet datasets. This training enables effective alignment of image and text representations within a shared embedding space. The VLM acts as the model’s ‘brain,’ processing instructions and current visual input to understand the task state. Subsequently, an ‘action expert’ module translates the VLM’s state information into robot actions. This work builds upon DexVLA [46], which employs a 2 billion parameter Qwen2-VL [2, 43] VLM as its backbone and a 1 billion parameter ScaleDP [57] (a diffusion policy variant) as its action expert. DexVLA undergoes three training stages: a 100-hour cross-embodiment training phase (Stage 1) followed by embodiment-specific training (Stage 2), and an optional task-specific training (Stage 3) for complex tasks. All three stages utilize 2D visual input. While these VLA models exhibit impressive capabilities in diverse manipulation tasks, their reliance on 2D vision limits their performance in tasks requiring 3D understanding, such as object deception via photographs or generalization across varying table heights. The following section aim to illustrate how to inject the 3D world into the pre-trained VLA. The overall framework can be viewed at Figure 2.

### 3.2. Injecting Point Cloud into VLA

**Motivation.** As previously established, Vision-Language-Action (VLA) models are typically pre-trained on large-scale 2D robotic datasets. A critical observation underpinning our approach is the inherent disparity in data scale between existing 2D pretraining corpora and emerging 3D robotic datasets. Specifically, we posit that 3D sensor data (e.g., point clouds, depth maps) remains orders of magnitude smaller in volume compared to 2D vision-language datasets, a consequence of the extensive historical focus on 2D perception in robotics research. This discrepancy necessitates a methodology that preserves the rich visual representations learned from 2D pretraining while effectively integrating sparse 3D data.

A naive strategy to address this challenge involves directly converting 3D visual input into 3D visual tokens and blending them into the Large Language Model (LLM) - a popular approach that has been leveraged by many 3DVLM, such as LLaVA-3D [56]. However, current vision-language models exhibit limited 3D comprehension when fine-tuned on small-scale 3D datasets, a limitation exacerbated by two factors: (1) the substantial domain gap between 2D pixels and 3D geometric structures, and (2) the scarcity of high-

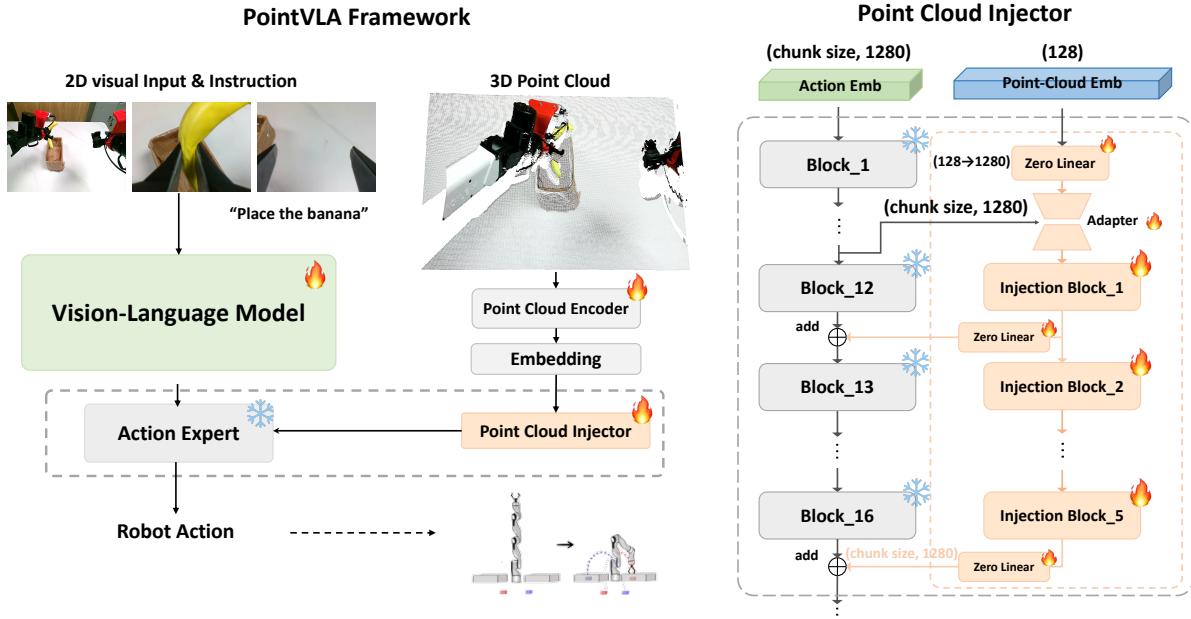


Figure 2. **Overview of PointVLA framework.** **Left:** The 2D image observation and instruction are processed by the vision-language model. The vanilla action expert remains frozen, while the new point cloud representation is integrated into the action expert through a modular network. **Right:** Details of the point cloud injector.

quality 3D-text paired data compared to the abundance of image-text and text-only corpora. To circumvent these issues, we propose a paradigm that treats 3D point cloud data as a complementary conditioning signal rather than a primary input modality. This strategy decouples 3D processing from the core 2D visual encoder, thereby preserving the integrity of pretrained 2D representations while enabling the model to leverage geometric cues. By design, our approach mitigates catastrophic forgetting of 2D knowledge and reduces the risk of overfitting to limited 3D data.

**Model architecture for point cloud injector.** The overall architecture of the point cloud injector is shown in Figure 1 (right). Specifically, for the incoming point cloud embedding, we first transform the channel dimension to match that of the vanilla action expert. Since the action embedding from the point cloud can be large, depending on the chunk size, we design an action embedding bottleneck to compress the information from the action expert while aligning it with the 3D point cloud embedding. For selected blocks in the action expert, we first apply an MLP layer as an adapter for each block, followed by an addition operation to inject the point cloud embedding into the model.

Notably, we avoided injecting the 3D features into every block of the action expert for two primary reasons. First, the computational cost would be prohibitively high due to the required conditioning blocks. Second, injection inevitably alters the model’s representation at the affected block. Given that we aimed to minimize interference

from the limited 3D visual knowledge on the pre-trained action embedding derived from 2D visual input, we conducted an analysis to identify blocks that could be skipped during inference without compromising performance. We subsequently injected 3D features only into these less critical blocks.

**Point cloud encoder.** Consistent with observations in DP3 [51] and iDP3 [50], we found that pre-trained 3D visual encoders hindered performance, often preventing successful robot behavior learning in new environments. Therefore, we adopted a simplified, hierarchical convolutional architecture. The upper convolutional layers extracted low-level features, while the lower convolutional blocks learned high-level scene representations. The max pooling was employed between layers to progressively reduce point cloud density. Finally, we concatenated feature embeddings from each convolutional block into a unified embedding, encapsulating multi-level 3D representation knowledge. The extracted point cloud feature embedding is retained for subsequent use. This architecture is similar to the iDP3 encoder. We believe that employing a more advanced point cloud encoder could further enhance model performance. However, as this is not the core novelty of our approach, we leave it for future discussion.

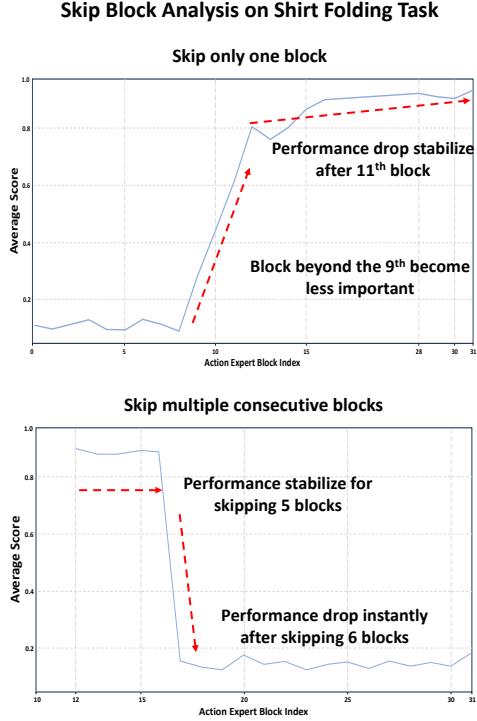


Figure 3. Skip block analysis for action expert in VLA model. **Left:** skipping only one block at a time. **Right:** Skipping multiple consecutive blocks starting from the 11th block.

### 3.3. Which Blocks to Inject Point Cloud? A Skip Block Analysis

As mentioned earlier, injecting the point cloud into every block of the action expert is not ideal, as it increases computational cost and disrupts the vanilla action representation learned from extensive 2D visual-based robot data. Therefore, we analyze which blocks in the action expert are less critical—i.e., those that can be skipped during inference without affecting performance. This approach aligns conceptually with techniques used in image generation, vision models, and large language models [10, 18, 26, 38]. Specifically, we use shirt folding in DexVLA [46] as a case study for our analysis. Recall that the DexVLA is equipped with 1 billion parameter action expert with 32 diffusion transformer blocks. The evaluation follows the same metrics—average score, a standard measure for long-horizon tasks [4, 31, 46]—by dividing the task into multiple steps and assessing performance based on step completion. We start by skipping one block at a time and summarize our findings in the figure below.

We illustrated the results in Figure 3 (left). Our experiment reveals that the first 11 blocks are crucial for the model—skipping any of them leads to a significant drop in performance. Specifically, when blocks before layer 11 are

skipped, the gripper fails to close tightly, making it difficult for the model to complete the task. However, starting from the 11 block onward, skipping a single block becomes acceptable, up until the last block. This indicates that blocks 11 to 31 contribute less to performance after training. To further investigate which blocks would be fit for point cloud injection, we conduct a multi-block skipping analysis from block 11 onward, as shown in Figure 3 (right). We find that up to five consecutive blocks can be skipped before the model fails at task completion. This suggests that 3D representation can be selectively injected into the action expert through specific blocks, optimizing efficiency without significantly impacting performance. As a result, we set all 3D conditioning blocks to be trainable when new data is introduced. We freeze all modules in the vanilla action expert except for the final layers, which are adjusted to fit the embodiment’s output. Ultimately, we only train five additional injection blocks, which are lightweight and fast during inference, making our approach highly cost-efficient.

## 4. Experiment

This section presents a series of experiments to validate the effectiveness of our approach. Specifically, we provide detailed descriptions of the real robot and workspace setup in Section 4.1. We evaluate the few-shot multi-tasking setting and long-horizon challenging tasks in Sections 4.2 and 4.3, respectively. In Sections 4.4 and 4.5, we explore two unique types of generalization that can only be observed with 3D visual input. Finally, we compare our method against simulation benchmarks.

### 4.1. Implementation Details

In this work, we conduct real robot experiments on two embodiments:

- **Bimanual UR5e.** Two UR5e robots, each with a RoboTiq parallel jaw gripper and a wrist-mounted camera. A top-down camera is positioned between the two arms. This setup has a total of three camera views and a 14-dimensional configuration and action space. Data is collected at 15Hz. We use the RealSense D435i camera as the wrist camera.
- **Bimanual AgileX.** Two 6-DoF AgileX arms, each with a wrist-mounted camera and a base camera. This setup has a 14-dimensional configuration and action space, supported by three cameras in total. Data is collected at 30Hz. We use the RealSense D435i camera as a wrist camera.

We use RealSense L515 camera to collect the point cloud. We set the VLM model trainable as the model needs to learn new language instruction. For both experiments, we use stage 1 pre-trained weights from DexVLA [46] and fine-tune for our model. We use the same training hyperparameters as the stage 2 training in DexVLA, and use the

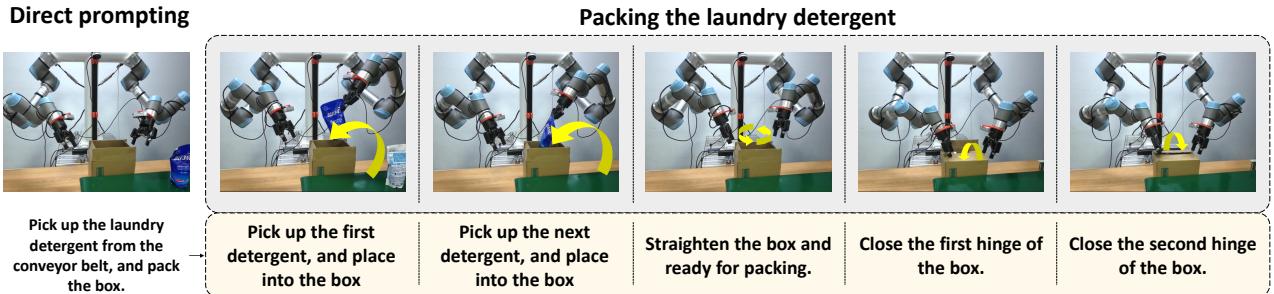


Figure 4. **Setup for bimanual UR5e.** We utilize three cameras: two RealSense D435i mounted on the wrists and one RealSense L515 positioned above. Our model is evaluated on a challenging long-horizon task that involves picking up two laundry detergent bottles from a moving conveyor belt and packing them into a box.

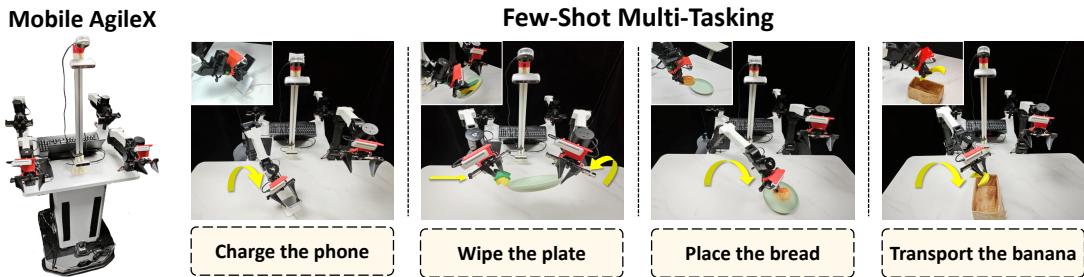


Figure 5. **Setup for bimanual AgileX.** We utilize three cameras: two RealSense D435i mounted on the wrists and one RealSense L515 positioned above. Our model is evaluated on four tasks in a few-shot setting.

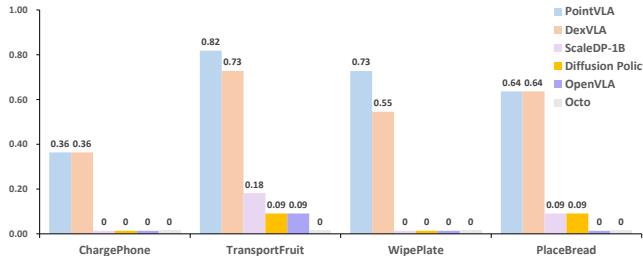


Figure 6. **Experimental results on few-shot multi-tasking on bimanual AgileX.**

last checkpoint for evaluation to avoid cherry picking. We set chunk size to 50 for all tasks.

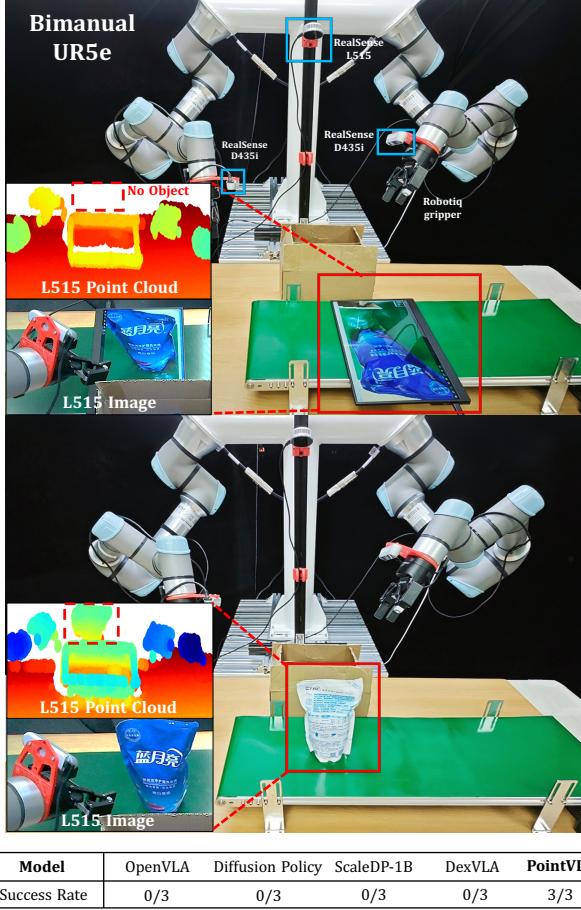
**Baseline.** In our experiments, we compared with many state-of-the-art model, including the Diffusion Policy (DP) [9], 3D Diffusion Policy (DP3) [51], ScaleDP-1B [57], a variant of scaling up diffusion policy to 1B parameters, Octo [34], OpenVLA [25], and DexVLA [46]. Note that since PointVLA is built on top of DexVLA, the DexVLA can be viewed as an ablation of our proposed PointVLA without the incorporation of 3D point cloud data.

## 4.2. Few-Shot Multi-Tasking

**Task description.** As illustrated in Fig 5, we designed four few-shot tasks for our real-world experiment: **ChargePhone**, **WipePlate**, **PlaceBread**, **TransportFruit**. Objects were placed randomly within a small range, and we report the average success rate for each method. **1) ChargePhone:** The robot picks up a smartphone and places it on a wireless charger. The phone’s size tests action precision, while its fragility requires careful handling. **2) WipePlate:** The robot simultaneously picks up a sponge and a plate, using the sponge to wipe the plate, assessing bimanual manipulation skills. **3) PlaceBread:** The robot picks up a piece of bread and places it on a plate. A thin foam layer under the bread ensures height generalization testing. **4) TransportFruit:** The robot picks up a randomly oriented banana and places it in a centrally located box.

As our approach aims to verify the few-shot multi-tasking ability of the model, we collected 20 demonstrations for each task, resulting in a total of 80 demonstrations. The objects’ positions are randomized within a small space. These tasks assess the model’s capability to manage both independent and coordinated robot movements across diverse scenarios. All data are collected using 30Hz.

**Experimental results.** The experimental results are pre-



**Figure 7. The experimental setup for real-vs-photo discrimination.** We replace the real laundry detergent with its photo displayed on a screen placed on the conveyor belt. From an exocentric view, the image appears noticeably different from the real object. However, from the egocentric top camera’s perspective, the photo closely resembles the actual detergent. We show experimental results on the bottom table.

sented in Table 6, where our method outperforms all baselines in this scenario. Notably, the Diffusion Policy fails in most cases, likely because the sample size for each task is too small, causing the action representation space to become entangled—an observation consistent with previous findings in the previous literature [47]. Furthermore, even increasing the model size (ScaleDP-1B) does not lead to significant improvement.

DexVLA demonstrates strong few-shot learning capabilities despite the limited data; however, its performance remains on par with or inferior to PointVLA. The incorporation of point cloud data in PointVLA enables more sample-efficient learning, emphasizing the necessity of integrating 3D information into the model. More importantly, our results confirm that our method successfully preserves the

**Table 1. Experimental results for long-horizon task on bimanual UR5e.** The task is completed in a sequence. The Avg. Len. denotes the average success length of the model.

Method	Packing Laundry Detergent					Avg. Len.
	1	2	3	4	5	
Octo [34]	1/11	1/11	0	0	0	0.27
OpenVLA [24]	2/11	1/11	0	0	0	0.36
Diffusion Policy [9]	2/11	1/11	0	0	0	0.36
ScaleDP-1B [57]	4/11	2/11	0	0	0	0.72
DexVLA [46]	2/11	5/11	1/11	1/11	0	1.72
<b>PointVLA</b>	3/11	1/11	1/11	2/11	2/11	2.36

ability to learn from the 2D pre-trained VLA.

### 4.3. Long-Horizon Task: Packing on Assembly Line

Beyond conventional multi-tasking, we further fine-tune PointVLA on long-horizon packing tasks, as illustrated in Figure 4. This is an exceptionally challenging task for several reasons. First, the assembly line is in motion, requiring the robot to quickly and precisely grasp objects. Second, the embodiment in this scenario differs from those present in the pre-trained data, necessitating rapid adaptation to a completely new setup. Third, as a long-range task, the robot must sequentially pick and place two bags of laundry detergent before sealing the packing box. These complexities make the task highly demanding.

The evaluation metrics are detailed in the Appendix. As shown in Table 1, PointVLA achieves the highest average length in long-range tasks, surpassing DexVLA, a strong baseline, by 0.64. It also outperforms several other baselines. However, the next section highlights an even more critical aspect of PointVLA, the object hallucination problem.

### 4.4. Real-vs-Photo Discrimination

In this section, we explore a unique setup called real-vs-photo discrimination. Specifically, we replace the real object with a picture of the object. From a 2D perspective, the “fake” object displayed on a screen appears nearly identical to the real object, yet it does not physically exist. Humans can easily recognize this discrepancy and refrain from reaching for the object because we understand it is not real—but can a robot model do the same?

To illustrate this scenario, we conduct an experiment using a bimanual UR5e in a packing task. We set the experiments by replacing laundry detergent with a photo of the laundry detergent projected on a screen. The experimental setup can be found in Figure 7. From an exocentric view, the image appears noticeably different from the real object. However, from the egocentric top camera’s perspective, the photo closely resembles the actual detergent. We observe that conventional 2D-based vision-language-action models, such as OpenVLA and DexVLA, fail to distinguish

between the image and a real object. These models attempt to grasp the object, and in the case of DexVLA, it repeatedly tries to pick up the non-existent detergent. Since the model believes the object is present but continuously fails to grasp it, it enters a repetitive grasping loop. In contrast, PointVLA successfully recognizes that no real object exists on the conveyor belt. By leveraging 3D spatial understanding, it determines that the space where the object should be is actually empty. This key advantage highlights the strength of our approach, demonstrating the superiority of 3D-aware models in mitigating object hallucination.

#### 4.5. Height Adaptability

In our context, height generalization refers to a model’s ability to adapt to different table heights. This is a crucial capability for robotic models, as most demonstrations are conducted on a fixed table height. However, what happens when the robot is deployed in an environment with a significantly different table height than what it was trained on?

To investigate this question, we designed an experiment illustrated in Figure 8. Specifically, in the “place bread” task, we placed a foam plastic layer beneath the bread. During training, this foam layer was 3mm thick, and all collected data was based on this height. During inference, we increased the foam thickness to 52mm to evaluate the model’s height generalization. Our observations show that conventional 2D-based VLA models, such as OpenVLA [25], DP [9], ScaleDP-1B [57], and DexVLA [46] all failed in this scenario. Upon detecting the bread, these models attempted to push it down and grasp it at the height seen in the training data, failing to adjust to the increased height. In contrast, PointVLA successfully completed the task. By leveraging the point cloud data, it accurately perceived the bread’s new height, adjusted its gripper accordingly, and executed a successful pick-up. This experiment demonstrates that incorporating 3D information enables VLA models to handle variations in object height—an ability that purely 2D-based models lack.

#### 4.6. Experimental Results on Simulation Benchmarks

We evaluate our approach on RoboTwin [32], a widely used mobile bimanual platform equipped with 14-degree-of-freedom robots. This benchmark encompasses a diverse set of tasks. We compare our method against Diffusion Policy [9] and 3D Diffusion Policy (DP3) [51]. Diffusion Policy is a well-established baseline for visuomotor policy learning, while DP3 extends it to the 3D domain. The vanilla DP3 uses only point cloud data as input. For a fair comparison, we also incorporate RGB images into DP3. The implementation is carried out by RoboTwin. We compare both versions of DP3 alongside the vanilla Diffusion Policy. In all experiments, we use a standard image resolution of  $320 \times 180$  for camera input, including the L515 and the top camera.

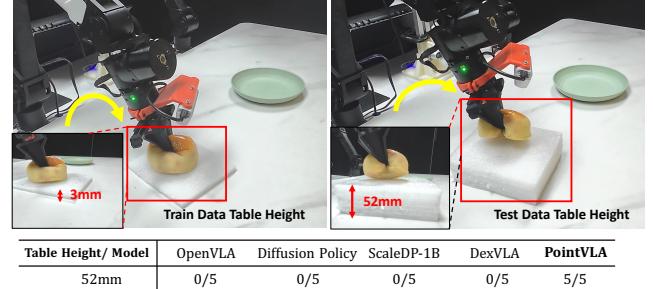


Figure 8. **Height adaptability for PointVLA.** Even if the model is trained on a standard height, when PointVLA encounters different table heights at test time, it can adapt to the new height and successfully complete the task. In contrast, conventional 2D vision-based imitation learning methods fail completely.

tion of  $320 \times 180$  for camera input, including the L515 and the top camera.

The tests were conducted using datasets of 20 and 50 samples. Following the training setup in RoboTwin, the policy was trained using three random seeds (0, 1, 2) without cherry picking for each experiment. Each policy was then tested 100 times, yielding three success rates. The mean and standard deviation of these success rates were computed to obtain the experimental results presented below.

The experimental results are presented in Table 2. Baseline results, including those for the 3D Diffusion Policy and Diffusion Policy, are reported by RoboTwin. Notably, across all tasks and diverse settings, our proposed PointVLA achieves the highest average success rate, regardless of whether it is trained on 20 or 50 demonstrations. This demonstrates that our method is effective even when data resources are limited and continues to perform well when abundant training data is available.

Additionally, we observe that for purely 3D models like DP3, directly incorporating RGB input can negatively impact performance. In contrast, our approach highlights the necessity of conditionally integrating 3D point cloud data into the model, which significantly enhances performance compared to models that rely solely on 2D visual input.

## 5. Conclusion

While Vision-Language-Action (VLA) models excel in robot learning through large-scale 2D pretraining, their reliance on RGB inputs limits 3D spatial reasoning. Retraining with 3D data is costly, and discarding 2D datasets reduces generalization. To address this, we introduced PointVLA, a framework that enhances pre-trained VLAs with 3D point cloud inputs while preserving their 2D representations. By integrating a modular 3D feature injector and leveraging skip block analysis, our method incorporates spatial information efficiently without full retrain-

Task	Number of Demonstrations		Task	Number of Demonstrations	
	20	50		20	50
<i>Block Hammer Beat</i>		<i>Block Handover</i>			
DP3 (Point Cloud)	47.7 ± 7.4	58.3 ± 6.5	DP3 (Point Cloud)	82.7 ± 6.1	85.0 ± 15.6
DP3 (Point Cloud+RGB)	44.7 ± 3.8	79.0 ± 2.0	DP3 (Point Cloud+RGB)	88.7 ± 5.0	94.3 ± 7.2
DP	0.0 ± 0.0	0.0 ± 0.0	DP	0.0 ± 0.0	0.0 ± 0.0
PointVLA	61.2 ± 4.8	84.6 ± 3.9	PointVLA	89.7 ± 4.3	96.1 ± 1.9
<i>Blocks Stack (Easy)</i>		<i>Blocks Stack (Hard)</i>			
DP3 (Point Cloud)	3.3 ± 3.2	17.0 ± 7.0	DP3 (Point Cloud)	0.3 ± 0.6	1.7 ± 0.6
DP3 (Point Cloud+RGB)	2.7 ± 1.2	17.0 ± 1.0	DP3 (Point Cloud+RGB)	0.0 ± 0.0	2.0 ± 2.0
DP	0.0 ± 0.0	0.0 ± 0.0	DP	0.0 ± 0.0	0.0 ± 0.0
PointVLA	10.8 ± 2.8	24.3 ± 4.7	PointVLA	2.3 ± 0.9	3.7 ± 1.4
<i>Bottle Adjust</i>		<i>Container Place</i>			
DP3 (Point Cloud)	55.7 ± 1.5	70.7 ± 2.5	DP3 (Point Cloud)	52.7 ± 4.5	74.0 ± 5.6
DP3 (Point Cloud+RGB)	28.3 ± 12.9	27.7 ± 16.5	DP3 (Point Cloud+RGB)	38.0 ± 7.9	58.3 ± 5.9
DP	13.0 ± 11.8	24.7 ± 13.8	DP	5.3 ± 4.2	16.3 ± 2.5
PointVLA	61.2 ± 2.8	74.5 ± 9.7	PointVLA	58.5 ± 4.8	81.3 ± 3.9
<i>Diverse Bottles Pick</i>		<i>Dual Bottles Pick (Easy)</i>			
DP3 (Point Cloud)	13.3 ± 5.5	34.7 ± 6.7	DP3 (Point Cloud)	37.0 ± 4.6	60.3 ± 7.1
DP3 (Point Cloud+RGB)	0.7 ± 0.6	5.3 ± 2.1	DP3 (Point Cloud+RGB)	29.7 ± 3.5	67.3 ± 9.3
DP	0.0 ± 0.0	0.3 ± 0.6	DP	1.3 ± 1.5	26.7 ± 3.1
PointVLA	21.5 ± 3.8	38.7 ± 5.7	PointVLA	41.2 ± 3.2	68.5 ± 4.8
<i>Dual Bottles Pick (Hard)</i>		<i>Dual Shoes Place</i>			
DP3 (Point Cloud)	33.0 ± 2.6	48.0 ± 5.2	DP3 (Point Cloud)	5.7 ± 0.6	10.0 ± 2.6
DP3 (Point Cloud+RGB)	23.0 ± 2.0	46.3 ± 7.8	DP3 (Point Cloud+RGB)	1.7 ± 2.9	3.7 ± 0.6
DP	2.0 ± 1.7	32.3 ± 5.9	DP	0.0 ± 0.0	3.0 ± 1.7
PointVLA	34.2 ± 2.7	51.3 ± 3.2	PointVLA	7.5 ± 1.8	13.1 ± 2.9
<i>Empty Cup Place</i>		<i>Mug Hanging (Easy)</i>			
DP3 (Point Cloud)	33.0 ± 6.2	70.3 ± 7.2	DP3 (Point Cloud)	7.3 ± 2.9	14.0 ± 3.6
DP3 (Point Cloud+RGB)	26.3 ± 10.4	71.3 ± 4.0	DP3 (Point Cloud+RGB)	1.0 ± 1.0	2.0 ± 2.0
DP	0.3 ± 0.6	14.7 ± 6.0	DP	0.0 ± 0.0	0.0 ± 0.0
PointVLA	41.2 ± 4.8	73.5 ± 4.2	PointVLA	9.3 ± 1.9	19.1 ± 2.8
<i>Mug Hanging (Hard)</i>		<i>Pick Apple Messy</i>			
DP3 (Point Cloud)	12.7 ± 0.6	11.0 ± 6.1	DP3 (Point Cloud)	5.7 ± 4.5	10.7 ± 4.0
DP3 (Point Cloud+RGB)	0.0 ± 0.0	2.0 ± 2.0	DP3 (Point Cloud+RGB)	6.7 ± 2.3	28.7 ± 9.5
DP	0.0 ± 0.0	0.3 ± 0.6	DP	3.3 ± 1.5	6.0 ± 5.0
PointVLA	13.5 ± 1.7	16.2 ± 1.9	PointVLA	9.3 ± 2.8	31.5 ± 4.7

Table 2. **Experimental results on RoboTwin [32].** Performance comparison across different tasks and demonstrations (results for 20 and 50 demonstrations).

ing. Experiments in both simulated and real-world environments demonstrate PointVLA’s effectiveness, achieving few-shot multi-task learning (4 tasks with only 20 demonstrations each) and excelling in long-horizon tasks like dynamic item packing. Real-world tests on bimanual robots (UR5e and AgileX arms) further validate its practicality and safety. Our work highlights the feasibility of augmenting pre-trained robot models with new modalities without costly retraining. Future work includes expanding 3D-aware pretraining across larger datasets.

## References

- [1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. 3
- [2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 3
- [3] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. 2
- [4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky.  $\pi_0$ : A vision-language-action flow model for general robot control, 2024. 2, 3, 5
- [5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 2
- [6] Qingwen Bu, Guanghui Ren, Chimeng Liu, Chengen Xie, Modi Shi, Xindong He, Jianheng Song, Yuxiang Lu, Siyuan Feng, Yao Mu, Chengyue Zhao, Shukai Yang, Ziyu Xiong, Xu Huang, Dafeng Wei, Guo Xu, Yi Liu, Yuxin Jiang, Xiuqi Cui, Cheng Ruan, Jia Zeng, Lei Yang, Li Chen, Yan Ding, Jisong Cai, Chonghao Sima, Huijie Wang, Yongjian Shen, Jialu Li, Jiaqi Zhao, Jianchao Zhu, Jiaqi Shan, Yuehan Niu, Cheng Jing, Mingkang Shi, Chi Zhang, Qinglin Zhang, Cunbiao Yang, Wenhao Wang, Xuan Hu, Ping Luo, Junchi Yan, Bin Zhao, Jiangmiao Pang, Jianlan Luo, Hongyang Li, Yu Qiao, and Maoqing Yao. Agibot world colosseo: Large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:25xx.xxx-to-appear*, 2025. 2
- [7] 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
- [8] Zhe Chen, Jiannan Wu, Wenhui Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024. 2, 3
- [9] 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. 1, 3, 6, 7, 8
- [10] Javier Martín Daniel Verdú. Flux.1 lite: Distilling flux1.dev for efficient text-to-image generation. 2024. 5
- [11] Sudeep Dasari, Oier Mees, Sebastian Zhao, Mohan Kumar Srirama, and Sergey Levine. The ingredients for robotic diffusion transformers. *arXiv preprint arXiv:2410.10088*, 2024. 3
- [12] Pengxiang Ding, Han Zhao, Wenjie Zhang, Wenxuan Song, Min Zhang, Siteng Huang, Ningxi Yang, and Donglin Wang. Quar-vla: Vision-language-action model for quadruped robots. pages 352–367. Springer, 2024. 3
- [13] Pengxiang Ding, Jianfei Ma, Xinyang Tong, Binghong Zou, Xinxin Luo, Yiguo Fan, Ting Wang, Hongchao Lu, Panzhong Mo, Jinxin Liu, et al. Humanoid-vla: Towards universal humanoid control with visual integration. *arXiv preprint arXiv:2502.14795*, 2025. 3
- [14] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu.

- Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. *arXiv preprint arXiv:2307.00595*, 2023. 3
- [15] 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. 3
- [16] 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.
- [17] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puha Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. In *ICLR 2024 Workshop: How Far Are We From AGI*. 3
- [18] Ajay Jaiswal, Bodun Hu, Lu Yin, Yeonju Ro, Shiwei Liu, Tianlong Chen, and Aditya Akella. Ffn-skipllm: A hidden gem for autoregressive decoding with adaptive feed forward skipping. *arXiv preprint arXiv:2404.03865*, 2024. 5
- [19] Yueru Jia, Jiaming Liu, Sixiang Chen, Chenyang Gu, Zhilue Wang, Longzan Luo, Lily Lee, Pengwei Wang, Zhongyuan Wang, Renrui Zhang, et al. Lift3d foundation policy: Lifting 2d large-scale pretrained models for robust 3d robotic manipulation. *arXiv preprint arXiv:2411.18623*, 2024. 3
- [20] Siddharth Karamchetti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. *arXiv preprint arXiv:2402.07865*, 2024. 2, 3
- [21] Rituraj Kaushik, Timothée Anne, and Jean-Baptiste Mouret. Fast online adaptation in robotics through meta-learning embeddings of simulated priors. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5269–5276. IEEE, 2020. 2
- [22] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024. 3
- [23] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamchetti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yun-liang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 2, 3
- [24] Moo Jin Kim, Karl Pertsch, Siddharth Karamchetti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 3, 7
- [25] Moo Jin Kim, Karl Pertsch, Siddharth Karamchetti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. 1, 2, 6, 8
- [26] Black Forest Labs. Flux. <https://github.com/black-forest-labs/flux>, 2024. 5
- [27] Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation, 2024. 3
- [28] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 3
- [29] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [30] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhusu Li, Hao Yang, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024. 2, 3
- [31] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022. 5
- [32] Yao Mu, Tianxing Chen, Shijia Peng, Zanxin Chen, Zeyu Gao, Yude Zou, Lunkai Lin, Zhiqiang Xie, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). *arXiv preprint arXiv:2409.02920*, 2024. 2, 8, 9
- [33] Dantong Niu, Yuvan Sharma, Giscard Biambry, Jerome Quenum, Yutong Bai, Baifeng Shi, Trevor Darrell, and Roei Herzig. Llarva: Vision-action instruction tuning enhances robot learning. *arXiv preprint arXiv:2406.11815*, 2024. 3
- [34] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024. 6, 7
- [35] 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. 2024. 2, 3
- [36] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025. 3
- [37] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025. 3
- [38] Zhiqiang Shen, Zechun Liu, and Eric Xing. Sliced recursive transformer. In *European Conference on Computer Vision*, pages 727–744. Springer, 2022. 5
- [39] Ishika Singh, Anthony Liang, Mohit Shridhar, and Jesse Thomason. Self-supervised 3d representation learning for robotics. In *ICRA2023 Workshop on Pretraining for Robotics (PT4R)*, 2023. 3

- [40] Xinyang Tong, Pengxiang Ding, Donglin Wang, Wenjie Zhang, Can Cui, Mingyang Sun, Yiguo Fan, Han Zhao, Hongyin Zhang, Yonghao Dang, et al. Quartonline: Latency-free large multimodal language model for quadruped robot learning. *arXiv preprint arXiv:2412.15576*, 2024. 3
- [41] 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. 3
- [42] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 2, 3
- [43] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 3
- [44] Taiyi Wang, Zhihao Wu, Jianheng Liu, Jianye Hao, Jun Wang, and Kun Shao. Distril: An asynchronous distributed reinforcement learning framework for on-device control agents. *arXiv preprint arXiv:2410.14803*, 2024. 3
- [45] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yixin Peng, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2409.12514*, 2024. 2, 3
- [46] Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 2025. 1, 2, 3, 5, 6, 7, 8
- [47] Kun Wu, Yichen Zhu, Jinming Li, Junjie Wen, Ning Liu, Zhiyuan Xu, Qinru Qiu, and Jian Tang. Discrete policy: Learning disentangled action space for multi-task robotic manipulation. *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025. 7
- [48] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024. 3
- [49] 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. 3
- [50] 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. 3, 4
- [51] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024. 2, 3, 4, 6, 8
- [52] Mengtan Zhang, Yi Feng, Qijun Chen, and Rui Fan. Dcpid-depth: Explicitly infusing dense correspondence prior to unsupervised monocular depth estimation. *arXiv preprint arXiv:2405.16960*, 2024. 3
- [53] Han Zhao, Min Zhang, Wei Zhao, Pengxiang Ding, Siteng Huang, and Donglin Wang. Cobra: Extending mamba to multi-modal large language model for efficient inference. 2025. 3
- [54] Wei Zhao, Pengxiang Ding, Min Zhang, Zhefei Gong, Shuanghao Bai, Han Zhao, and Donglin Wang. Vlas: Vision-language-action model with speech instructions for customized robot manipulation. *arXiv preprint arXiv:2502.13508*, 2025. 3
- [55] Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Ran Cheng, Yixin Peng, Chaomin Shen, et al. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. *arXiv preprint arXiv:2502.14420*, 2025. 3
- [56] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. *arXiv preprint arXiv:2409.18125*, 2024. 3
- [57] Minjie Zhu, Yichen Zhu, Jinming Li, Junjie Wen, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yixin Peng, Feifei Feng, et al. Scaling diffusion policy in transformer to 1 billion parameters for robotic manipulation. *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025. 3, 6, 7, 8
- [58] Minjie Zhu, Yichen Zhu, Xin Liu, Ning Liu, Zhiyuan Xu, Chaomin Shen, Yixin Peng, Zhicai Ou, Feifei Feng, and Jian Tang. Miphia: A comprehensive overhaul of multimodal assistant with small language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025. 3
- [59] Yichen Zhu, Minjie Zhu, Ning Liu, Zhiyuan Xu, and Yixin Peng. Llava-phi: Efficient multi-modal assistant with small language model. In *Proceedings of the 1st International Workshop on Efficient Multimedia Computing under Limited*, pages 18–22, 2024. 3