Grounding Actions in Camera Space: Observation-Centric Vision-Language-Action Policy

Tianyi Zhang[1, 2*], **Haonan Duan**[3*], **Haoran Hao**[4, 2], **Yu Qiao**[2], **Jifeng Dai**[5], **Zhi Hou**[2†]
[1]College of Computer Science and Technology, Zhejiang University
[2]Shanghai Artificial Intelligence Laboratory
[3]SenseTime Research
[4]School of Artificial Intelligence, Nanjing University
[5]Department of Electronic Engineering, Tsinghua University
tianyizhang0213@zju.edu.cn, duan.haonan10@gmail.com, houzhi91@gmail.com

Abstract

Vision-Language-Action (VLA) models frequently encounter challenges in generalizing to real-world environments due to inherent discrepancies between observation and action spaces. Although training data are collected from diverse camera perspectives, the models typically predict end-effector poses within the robot base coordinate frame, resulting in spatial inconsistencies. To mitigate this limitation, we introduce the Observation-Centric VLA (OC-VLA) framework, which grounds action predictions directly in the camera observation space. Leveraging the camera's extrinsic calibration matrix, OC-VLA transforms end-effector poses from the robot base coordinate system into the camera coordinate system, thereby unifying prediction targets across heterogeneous viewpoints. This lightweight, plug-and-play strategy ensures robust alignment between perception and action, substantially improving model resilience to camera viewpoint variations. The proposed approach is readily compatible with existing VLA architectures, requiring no substantial modifications. Comprehensive evaluations on both simulated and real-world robotic manipulation tasks demonstrate that OC-VLA accelerates convergence, enhances task success rates, and improves cross-view generalization.

Code — https://github.com/ZTY0213/OC-VLA
Extended version — https://arxiv.org/pdf/2508.13103

Introduction

Inspired by the remarkable progress of multimodal large models, recent advances in vision-language-action (VLA) models (Brohan et al. 2023b; Kim et al. 2024; Team et al. 2024; Black et al. 2024; Hou et al. 2025) have focused on leveraging large-scale robot data from heterogeneous sources for pre-training, with the objective of enhancing generalization capabilities. Although this paradigm has achieved impressive performance across a variety of benchmarks, it remains fundamentally constrained by the intrinsic limitations of the robotics domain—namely, the relatively modest scale and high cost of data collection when compared to the web-scale corpora used in vision-language model (VLM) pre-training (O'Neill et al. 2024; Walke et al. 2023). Consequently, the ability of current VLA models to generalize effectively in real-world environments remains limited, leaving substantial room for further advancement.

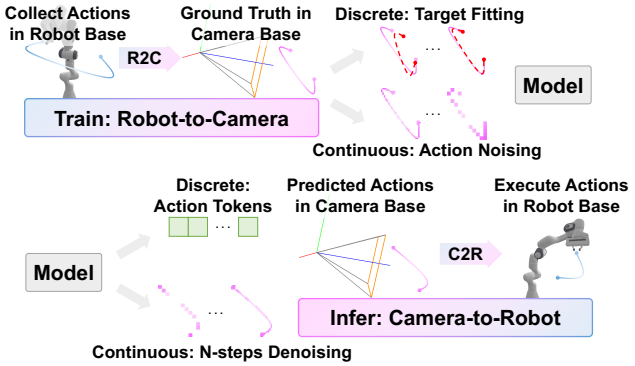

Figure 1: Full Pipeline of our method. We introduce OC-VLA framework, aligning the observation space and the prediction target with the camera extrinsic calibration matrix. It is simple and efficient, improve the performance of the VLA models without any extra GPU consumption.

A common practice in VLA modeling is to adapt pre-trained vision-language or vision encoders for downstream robotic tasks (O'Neill et al. 2024; Kim et al. 2024; Hou et al. 2025). However, these vision models are primarily trained and supervised within the image or camera coordinate system, resulting in latent representations that are inherently aligned with camera viewpoints. In contrast, most robotic control signals are defined in the robot base coordinate system (Brohan et al. 2023b; O'Neill et al. 2024; Kim et al. 2024; Hou et al. 2025). This discrepancy introduces a misalignment between the perception and action spaces, which can hinder effective policy learning, especially during the transfer of pretrained vision models to robotic control tasks.

Moreover, robot datasets are typically collected under diverse camera viewpoints and heterogeneous hardware configurations (O'Neill et al. 2024; Khazatsky et al. 2024; Walke et al. 2023), where the robot base is not always within

*These authors contributed equally.
†Corresponding Author

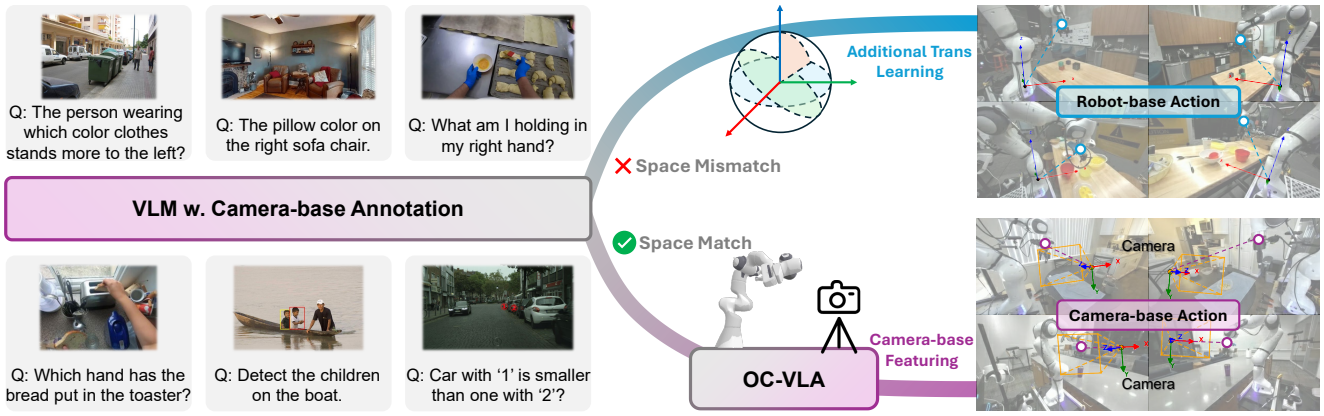

Figure 2: We introduce the Observation-Centric VLA (OC-VLA) framework. By transforming end-effector actions from the robot base coordinate to the third-person camera coordinate, OC-VLA aligns action predictions with visual observations across diverse viewpoints, enabling improved generalization and robustness in manipulation tasks.

the camera's field of view. In such settings, the same action expressed in the robot base coordinate system must be inferred from different third-person camera views. This implicitly requires the model to reconstruct or reason about consistent 3D actions from limited 2D observations—a fundamentally ill-posed challenge when only single- or dual-view inputs are available. Predicting actions defined in the robot base coordinate system becomes even more challenging, as it necessitates an implicit understanding of the transformation between robot and camera spaces. Such inconsistencies are particularly detrimental during large-scale pre-training (Brohan et al. 2023a; Kim et al. 2024), where diverse camera viewpoints are common: images capturing the same robot action from different angles are forced to share a single supervision signal in robot space, thereby introducing learning conflicts and hindering generalization.

To address these issues, we propose a novel paradigm that decouples the end-effector action from the robot base coordinate system and instead predicts actions directly in the third-person camera coordinate system, named Observation-Centric VLA (OC-VLA). Specifically, given the extrinsic transformation between the robot base and each camera, we transform the robot-space end-effector actions into their equivalent representations in the camera coordinate frame and adopt these as prediction targets. By anchoring the action target in the same space as the observation (i.e., the image plane), this formulation alleviates the misalignment between perception and action modalities and mitigates the ambiguity introduced by camera viewpoint variations. Furthermore, it explicitly encourages the model to learn the relative spatial relationships between the robot and the cameras, thereby enhancing its capacity to generalize effectively across diverse viewpoints and hardware configurations.

The proposed approach is evaluated across both simulated environments and real-world robotic platform. Experimental results consistently demonstrate that employing camera-space end-effector actions as prediction targets yields substantial performance gains over baselines that operate in robot coordinates. Notably, our method exhibits markedly improved adaptability to previously unseen camera viewpoints, underscoring its strong potential for robust generalization in diverse real-world deployment scenarios.

Related Work

Robotic Manipulation

Robotic manipulation still faces significant challenges in complex environments and tasks. Compared with traditional methods, learning-based manipulation has gained significant attention(Kroemer, Niekum, and Konidaris 2021). A common strategy for learning to predict actions is reinforcement learning (Dalal et al. 2024; Yamada et al. 2021; Xia et al. 2020).Another approach is to provide offline expert demonstrations for supervised learning (Brohan et al. 2023b; Shridhar, Manuelli, and Fox 2022, 2023). However, both approaches are data-driven and sensitive to environmental changes, limiting their effectiveness in open-world applications. Recently, the development of large language models (LLMs) and vision-language models (VLMs) has made reasoning and planning possible for solving complex tasks that require human knowledge (Li et al. 2024; Jin et al. 2024; Singh et al. 2023). However, due to limitations in their pre-training data, these models are still unable to control robots and address real-world tasks effectively. Vision-Language-Action (VLA) models (Brohan et al. 2023b; Kim et al. 2024; Hou et al. 2025; Cheang et al. 2024; Black et al. 2024) are trained on large-scale observation-action pairs and have strong capabilities in unified perception, reasoning, planning, and control, making them a promising solution for achieving unified robotic manipulation. Nevertheless, the generalization of current VLA models is limited, and the observation space action prediction is poorly investigated.

Vision-Language-Action Model

VLA models have become the popular framework for generalist robot policies. Recent advances leverage large-scale multi-modal backbones and foundation models (Wu et al. 2023; Cheang et al. 2024; Li et al. 2025; Huang et al. 2025;

Reuss et al. 2023; Ha, Florence, and Song 2023; Myers et al. 2023; Chen, Bahl, and Pathak 2023; Tian et al. 2024), improving generalization across tasks and embodiments. Diffusion models (Ho, Jain, and Abbeel 2020a; Rombach et al. 2022; Dhariwal and Nichol 2021; Peebles and Xie 2023) have shown strong performance in multi-modal action modeling (Wang et al. 2024; Wen et al. 2025), yet most existing approaches rely on U-Net or shallow cross-attention architectures, which limit scalability to more diverse tasks. To address complex scenarios, recent works integrate VLM embeddings with MLP diffusers (Team et al. 2024; Wen et al. 2024), or utilize Transformer-based (Vaswani et al. 2017) decoders for bimanual and multi-modal manipulation (Liu et al. 2024; Dasari et al. 2024), further pushing the frontier of unified VLA policy learning.

Method

In this section, we provide a detailed overview of OC-VLA, i.e., grounding actions in the camera space. We begin with the model structure and action modeling as preliminaries, followed by an introduction to the camera-centric action prediction approach. We then analyze the differences between camera-coordinate and robot-coordinate optimization.

Preliminary: Model Structure, Action Modeling

Vision-language-action (VLA) models have converged toward a common architectural pattern, where action prediction is built upon a vision-language backbone. Following this paradigm, we adopt a lightweight 334M VLA model (Hou et al. 2025) for evaluation, which has demonstrated competitive performance using only a third-person camera image and language instructions as input. Specifically, we follow Dita (Hou et al. 2025), where the language instruction is encoded using a CLIP text encoder (Radford et al. 2021), and the third-person image is processed using DINOv2 (Oquab et al. 2023). The image features are further selected and modulated by the language instruction via a Q-Former (Li et al. 2023) equipped with FiLM (Perez et al. 2018) conditioning layers.

Current VLA models typically employ one of two types of action spaces for end-effector control: discrete action spaces (Kim et al. 2024; Brohan et al. 2023b) and continuous action spaces (Team et al. 2024; Black et al. 2024). To thoroughly evaluate the effectiveness of our proposed approach, we conduct experiments on models using both types of action spaces. Based on the baseline architecture, we implement a variant specifically designed for discrete action prediction or continuous action prediction.

Observation-Centric Action Prediction

In current robotic datasets, action/pose annotations are often defined at a low level, either as joint commands or end-effector poses within the robot base coordinate frame. While these representations are widely used as supervision signals for Vision-Language-Action (VLA) models, they are tightly coupled with specific robot embodiment configurations, *rather than being derived from the observation space*. Consequently, it is difficult for the model to achieve a reasonable projection from image observation to corresponding actions, and thus the model generalization is limited, especially for novel camera views with a large variance from the seen camera views in the training set.

To ground actions in the observation space, it is necessary to first transform the actions from the robot (world) coordinate system into the camera coordinate system. We utilize the extrinsics of the camera to conduct the transformation. Specifically, given two nearby end-effector poses in the world coordinate frame, denoted as $\mathbf{P}_{\text{world}1} \in R^{4\times4}$ and $\mathbf{P}_{\text{world}2} \in R^{4\times4}$, where the matrix can be converted from a 3D rotation and a translation, the corresponding action can be derived accordingly,

$$\mathbf{A}_{\text{world}} = \mathbf{P}_{\text{world}2}{\mathbf{P}_{\text{world}1}}^{-1} \tag{1}$$

Meanwhile, we can get the corresponding poses in the camera coordinate as follows,

$$\mathbf{P}_{\text{cam}2} = \mathbf{T}\mathbf{P}_{\text{world}2}, \mathbf{P}_{\text{cam}1} = \mathbf{T}\mathbf{P}_{\text{world}1} \tag{2}$$

where $\mathbf{T} \in R^{4\times4}$ represents the world-to-camera transformation matrix, consisting of a 3D rotation and a translation. $\mathbf{P}$ represents the corresponding matrix. Then, we can obtain the corresponding actions in the camera space.

$$\mathbf{A}_{\text{cam}} = \mathbf{P}_{\text{cam}2}{\mathbf{P}_{\text{cam}1}}^{-1} \tag{3}$$

Lastly, we convert $\mathbf{A}_{\text{cam}} \in R^{4\times4}$ into the 7-dim actions $\langle \text{x}, \text{y}, \text{z}, \text{roll}, \text{pitch}, \text{yaw}, \text{gripper} \rangle$ for model optimization, where gripper is for the gripper position. Different from previous end-effector action prediction, the predicted action in our method is in the camera space.

During inference, we transform the actions in the camera space to robot coordinate space for robot control based on the camera calibration.

Analysis from Optimization Perspective

In this section, we provide a detailed analysis of the advantages of camera-centric action prediction. In details, we can get $\mathbf{A}_{\text{cam}}$ from equations 1, 2 and 3 as follow,

$$\mathbf{A}_{\text{cam}} = \mathbf{T}\mathbf{A}_{\text{world}}\mathbf{T}^{-1} \tag{4}$$

where $\mathbf{A}_{\text{cam}}$ is the camera-based action, $\mathbf{A}_{\text{world}}$ is the robot-based action, and $\mathbf{T}$ is the camera world-to-camera transformation matrix.

Meanwhile, given an end-effector pose $\mathbf{P}_{\text{world}}$ of the robot, we can get,

$$\mathbf{P}_{\text{cam}} = \mathbf{T}\mathbf{P}_{\text{world}} \tag{5}$$

Equations 4 and 5 present that both the end effector pose and action in world space require the transformation matrix $\mathbf{T}$ to be driven from representations in observation space.

In particular, the transformation matrix $\mathbf{T}$ varies across different robot setups. For instance, Droid (Khazatsky et al. 2024) features 1417 distinct camera viewpoints, requiring the model to internally infer the correct transformation $\mathbf{T}$ for each view to predict actions accurately in the robot's coordinate frame.

Besides, the traditional perception task is based on UV coordinates (image coordinates). According to the intrinsics of the camera, we can obtain the UV coordinate from $(X_{\text{cam}}, Y_{\text{cam}}, Z_{\text{cam}})$. Given that the intrinsic matrix $\mathbf{K}$, the image coordinates (u, v) can be calculated as:

$$u = \frac{f_x \cdot X_{\text{cam}}}{Z_{\text{cam}}} + c_x, \quad v = \frac{f_y \cdot Y_{\text{cam}}}{Z_{\text{cam}}} + c_y \tag{6}$$

Where f_x, f_y are the focal lengths in the x and y directions, c_x, c_y are the principal point coordinates (usually the image center). We observe that the camera coordinate can be directly derived from the UV coordinate, and the intrinsic parameters are usually consistent across cameras of the same model. However, translating a point from the camera coordinate system to the robot base coordinate requires the corresponding rotation matrix, which varies with different camera placements. As a result, learning this translation for robot space action prediction becomes more challenging due to the diversity in camera poses. *In contrast, observation-centric action prediction inherently avoids these issues, offering a more consistent mapping between observation and action.*

Experiments

In this section, we provide a detailed description of the pretraining data, followed by an overview of the model architecture for different action spaces. Next, we present the optimization process. Lastly, we present a comprehensive evaluation of the performance of our proposed method on both simulated benchmarks and real-world robotic platforms.

Pretraining Data

To ensure a comprehensive and fair evaluation of our proposed approach, we incorporate a pretraining stage in selected experiments. Pretraining provides the model with a stronger initialization, which is particularly beneficial when handling complex multimodal inputs and diverse visual contexts. Since our method operates from a third-person perspective and explicitly requires camera extrinsics to transform robot-centric actions into the camera frame, it is crucial to select a dataset that includes such calibration information.

For this purpose, we choose the Droid dataset (Khazatsky et al. 2024) for pretraining. This dataset consists of robotic manipulation trajectories captured from 1417 distinct third-person camera viewpoints, along with their corresponding extrinsic parameters, offering a wide range of visual perspectives and motion patterns. This diversity makes it an ideal choice for evaluating the generalizability and robustness of our observation-centric action prediction framework. Unless otherwise noted, all experiments involving pretrained models are initialized using weights obtained through pretraining on the Droid dataset (Khazatsky et al. 2024).

Model Details

In our experiments, we employ a typical lightweight VLM architecture, with distinct designs for the continuous and discrete action spaces. In the following, we detail the model implementations for each action space.

For the continuous action space model, we adopt a diffusion policy. In addition to language and image tokens, we concatenate the current timestep and the noise-perturbed action as inputs to the causal transformer. The entire transformer functions as a Diffusion Transformer (DiT) (Peebles and Xie 2023), which iteratively denoises the input over multiple steps to generate the final end-effector action.

For the discrete action space model, we pad zero vectors to align the action size after processing language and image inputs. The combined sequence is then fed into the Transformer. Despite using a causal mask during training, the model predicts the entire action sequence in a single pass, rather than autoregressively. This design enhances both semantic consistency across tokens and computational efficiency.

Optimization Details

The training objectives vary depending on the type of action space used. For models with a continuous action space, the objective is to minimize the mean squared error (MSE) between the robot's action (augmented with standard Gaussian noise) and the predicted noise, using DDPM (Ho, Jain, and Abbeel 2020b) with 100 timesteps. In contrast, for models with a discrete action space, the robot actions are normalized to a predefined range and quantized into discrete bins. The objective here is to minimize the cross-entropy loss between the predicted discrete actions and the ground-truth labels.

For diffusion evaluation, we use DDIM (Song, Meng, and Ermon 2021) with 10 timesteps during inference. The model is optimized using AdamW (Loshchilov and Hutter 2019) for 30,000 steps, with learning rates of $1e-4$ for both the causal Transformer and Q-Former, and $1e-5$ for DINOv2. Training is conducted with a batch size of 2048 across 8 NVIDIA A100 GPUs, with 256 samples per GPU. The model predicts actions in the third-person camera base coordinate, while the baseline model predicts actions in the robot base coordinate.

Simulation Evaluation

Simulation Dataset For simulated evaluation, we select ManiSkill2 (Gu et al. 2023) to assess the effectiveness and generalization capabilities of our proposed approach. ManiSkill2, the successor to the original SAPIEN ManiSkill (Mu et al. 2021) benchmark, has become a widely recognized and authoritative platform for evaluating the generalization performance of embodied agents in robotic manipulation. Meanwhile, ManiSkill2 includes 20 diverse task families, covering a broad range of real-world manipulation scenarios. Additionally, ManiSkill2 supports rendering observations from randomly sampled camera viewpoints, making it a suitable choice for our evaluation.

Setup To construct our benchmark, we select five representative tasks from the ManiSkill2 suite: PickCube-v0, StackCube-v0, PickSingleYCB-v0, PickClutterYCB-v0, and PickSingleEGAD-v0. We generate a pool of 300,000 randomly configured camera viewpoints. For each trajectory, 20 cameras are randomly sampled to render the demonstration, resulting in a dataset comprising over 40,000

Coord	Continuous	All	PickC	StackC	SingleYCB	ClutterYCB	EGAD
Robot	✓	45.2%	71.0%	62.0%	30.0%	15.0%	48.0%
Camera	✓	**53.2%**	**88.0%**	**65.0%**	**46.0%**	**19.0%**	48.0%
Robot	×	38.6%	61.0%	51.0%	28.0%	8.0%	45.0%
Camera	×	**52.4%**	**80.0%**	**65.0%**	**48.0%**	**19.0%**	**50.0%**

Table 1: Comparison on ManiSkill2 under Success rate. SingleYCB indicates PickSingleYCB, ClutterYCB indicates PickClutterYCB, SingleEGAD indicates PickSingleEGAD. Coord indicates the selected coordinate while training. Continuous indicates whether the action prediction is continuous or discrete.

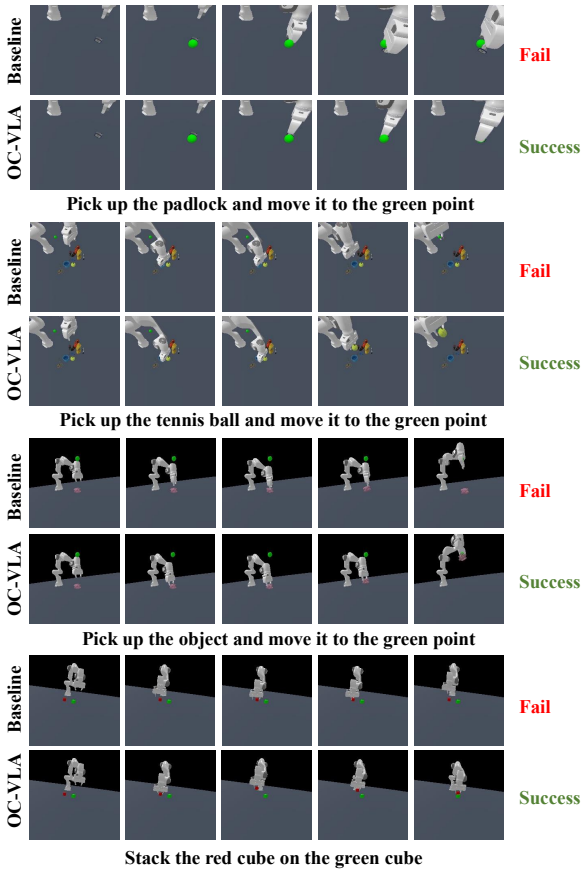

Figure 3: Qualitative Comparison on ManiSkill2 of OC-VLA and Baseline. OC-VLA show better performance on the grasp pose and searching for the goal point.

unique trajectories. We partition the generated data into training and validation sets using a 19:1 ratio. We make sure that each task family is represented in both sets, and that trajectories rendered from different camera viewpoints are distributed across the splits, thereby preventing data leakage. To address data imbalance, we replicate trajectories from underrepresented task families to equalize the number of samples across tasks during training. For closed-loop evaluation, we sample 100 trajectories from the validation set for each task family, resulting in an evaluation set of 500 trajectories. This evaluation benchmark is used to measure the success rate of the model across different manipulation tasks.

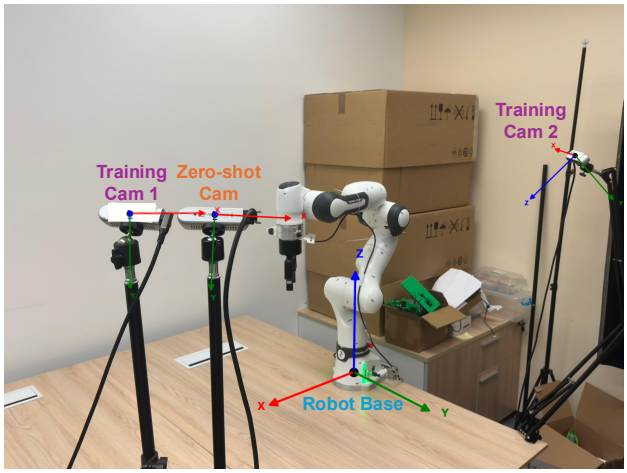

Figure 4: The real-world robot platform with a Franka Emika Panda robot, a Robotiq 2F-85 gripper and multiple RealSense D435i RGB-D cameras.

Comparisons Given the domain gap between Droid and Maniskill2, both the continuous and discrete action space models are trained from scratch in this evaluation. We conduct a comparative analysis of their performance under two supervision regimes: one using robot actions defined in the robot base coordinate frame, and the other using robot actions transformed into the third-person camera coordinate frame as the prediction targets. Figure 3 shows the qualitative results and Table 1 shows the quantitative results of the different models with different prediction target. The results demonstrate that, regardless of the type of action space used, employing robot actions defined in the third-person camera coordinate frame as prediction targets consistently improves task success rates. *This improvement is particularly pronounced in models utilizing a discrete action space, where we observe an increase in success rate of about 14%.*

Real Robot Evaluation

Setup We evaluate OC-VLA on a real-world Franka Robot setup, which comprises a 7-DoF Franka Emika Panda robot arm equipped with a Robotiq 2F-85 gripper as shown in Figure 4. Three RealSense D435i cameras are positioned to capture the environment from multiple third-person perspectives. Specifically, two cameras are used for both data collection and few-shot evaluation, while the remaining camera is reserved exclusively for zero-shot evaluation.

Method	Avg	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7
OpenVLA-OFT	63.3%	100.0%	80.0%	90.0%	80.0%	80.0%	80.0%	60.0%
OpenVLA-OFT (var)	42.0%	90.0%	70.0%	60.0%	40.0%	50.0%	50.0%	10.0%
π_0	50.7%	50.0%	70.0%	80.0%	60.0%	60.0%	70.0%	80.0%
π_0 (var)	34.7%	20.0%	40.0%	60.0%	40.0%	30.0%	30.0%	60.0%
Robot Base	58.0%	70.0%	70.0%	90.0%	60.0%	60.0%	60.0%	60.0%
Robot Base (var)	41.3%	40.0%	50.0%	70.0%	60.0%	40.0%	60.0%	60.0%
Camera Base (OC-VLA, ours)	**68.0%**	80.0%	80.0%	100.0%	80.0%	80.0%	70.0%	60.0%
Camera Base (var)	54.0%	70.0%	70.0%	100.0%	70.0%	60.0%	60.0%	70.0%

Method	Task 8	Task 9	Task 10	Task 11	Task 12	Task 13	Task 14	Task 15
OpenVLA-OFT	70.0%	50.0%	20.0%	20.0%	80.0%	50.0%	60.0%	30.0%
OpenVLA-OFT (var)	40.0%	50.0%	10.0%	10.0%	30.0%	50.0%	50.0%	20.0%
π_0	60.0%	40.0%	10.0%	20.0%	40.0%	50.0%	60.0%	10.0%
π_0 (var)	60.0%	30.0%	10.0%	10.0%	50.0%	40.0%	30.0%	10.0%
Robot Base	60.0%	60.0%	20.0%	40.0%	50.0%	60.0%	90.0%	20.0%
Robot Base (var)	30.0%	30.0%	10.0%	20.0%	30.0%	50.0%	50.0%	20.0%
Camera Base (OC-VLA, ours)	70.0%	60.0%	40.0%	50.0%	70.0%	60.0%	90.0%	30.0%
Camera Base (var)	40.0%	50.0%	20.0%	30.0%	20.0%	60.0%	70.0%	20.0%

Table 2: Quantitative results in Real robot experiments. Methods annotated with "(var)" indicate results obtained under zero-shot camera evaluation, while those without the annotation correspond to evaluations conducted using the Training Cam 1. Robot Base and Camera Base indicates the model we built in robot base coordinates and third-person camera base coordinate following Dita (Hou et al. 2025), respectively. Tasks samples and their corresponding simple descriptions is referred to Figure 5.

Method	Avg	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8
Robot Base (Fixed Camera)	66.3%	70.0%	70.0%	90.0%	60.0%	60.0%	60.0%	60.0%	60.0%
Cam Base (Fixed Camera)	77.5%	80.0%	80.0%	100.0%	80.0%	80.0%	70.0%	60.0%	70.0%
Robot Base (Camera Perturbations)	61.3%	80.0%	70.0%	50.0%	70.0%	40.0%	60.0%	50.0%	70.0%
Cam Base (Camera Perturbations)	73.8%	80.0%	80.0%	70.0%	90.0%	80.0%	60.0%	60.0%	70.0%

Table 3: Real robot experiments of different camera views. Fixed Camera means no camera perturbations while data collection. The meanings of the methods and the Task ID mappings follow the same convention as in Table 2.

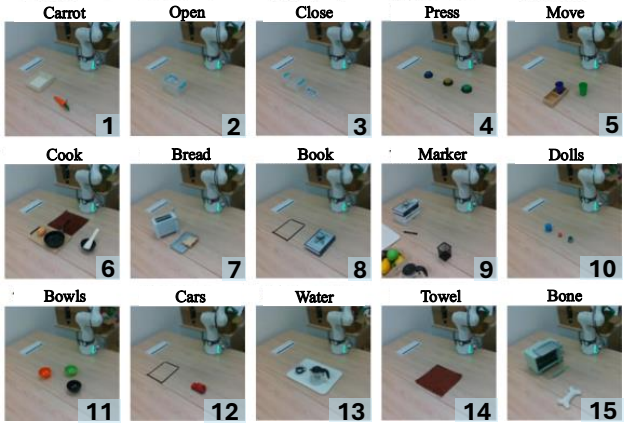

Figure 5: Task samples and description on the real-robot finetuning and evaluation

Data Collection and Model Finetuning We adopt a demonstration-based approach to collect two datasets from different viewpoints using Training Camera 1 and Training Camera 2, respectively. For the dataset collected with Camera 1, we record trajectories for 15 distinct tasks while keeping the camera position fixed throughout the entire data collection process. In contrast, the dataset collected with Camera 2 consists of trajectories for 8 tasks, during which we introduce slight perturbations to the camera position to simulate minor viewpoint variations. The collected tasks span a diverse set of categories, including pick & place, pouring, stacking, pick & rotation, pull & push, as well as other long-horizon tasks, aiming to comprehensively evaluate the true performance of the model. A detailed list of tasks is provided in the Extended Version. Following Dita (Hou et al. 2025), for each task in both datasets, we collect 10 demonstration trajectories, aiming to evaluate the model fitting ability under a 10-shot setting.

For model finetuning, we fine-tune the model pretrained on the Droid dataset, using either end effector actions defined in the third-person camera coordinate or those in the robot base coordinate as prediction targets. Both models are

optimized with AdamW (Loshchilov and Hutter 2019) for 20,000 steps with a batch size of 512. For a fair performance comparison, we also fine-tune the pretrained versions of OpenVLA-OFT (Kim et al. 2024), π_0 (Black et al. 2024) on our collected datasets, using their official training protocols. These models serve as baselines in our evaluation.

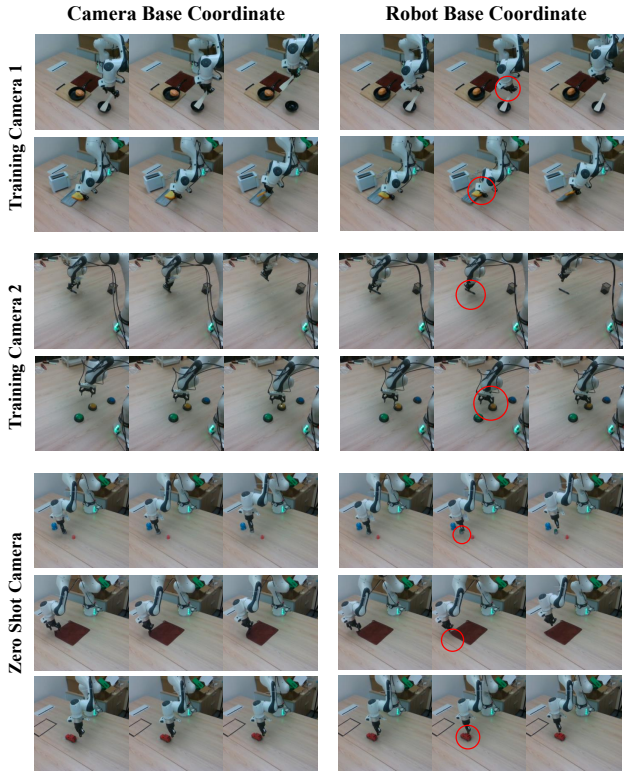

Figure 6: A qualitative comparison in real-robot experiments. Failures are highlighted with red circles.

Quantitative Evaluation and Comparison The evaluations are organized into the following three main settings:

- **Fixed Camera Viewpoint.** We fine-tune all models using 15 task demonstrations collected from Camera 1 and perform a unified evaluation. For each task, we conduct 10 trials and measure performance by computing the task success rate. In this setting, the camera viewpoint remains fixed and identical throughout both the fine-tuning and evaluation phases.
- **Slight Camera Perturbations** To further validate the robustness of our method, we introduce slight variations in the camera viewpoints. Specifically, we fine-tune the models using 8 task demonstrations collected from Camera 2, each exhibiting minor differences in camera placement. For evaluation, we position the camera in a similar configuration to the fine-tuning setup and recalibrate the camera to obtain updated extrinsic parameters. The camera remains fixed throughout the evaluation process.
- **Novel Camera Viewpoint.** To assess the model's robustness to changes in camera perspective, we conduct zero-shot evaluations using models fine-tuned with demonstrations from Camera 1. As illustrated in the Figure 4, we introduce a novel, previously unseen camera mounted near Camera 1, and perform all evaluations under this new fixed viewpoint without any additional fine-tuning.

Fix Camera View. As shown in the Table 2, under the 10-shot setting with a fixed camera viewpoint, the model fine-tuned using robot base coordinate actions already demonstrates competitive performance. However, when the prediction target is switched from robot-base coordinate actions to camera-base coordinate actions, the model achieves a further 10% improvement in the metric of success rate, surpassing the best-performing baseline, OpenVLA-OFT, fine-tuned on the same data. This indicates that our method can partially compensate for the limited pretraining data and model size by improving data efficiency.

Novel Camera View. For novel camera view, all models exhibit varying degrees of performance degradation in Table 2, as expected. Notably, OpenVLA-OFT, which performs as the best baseline under the 10-shot setting, suffers a performance drop of over 20%. In contrast, our method shows only a 14% decrease, outperforming all baselines in this setting. *These results highlight the added robustness to camera viewpoint changes when the model is trained to predict actions in the camera base coordinate frame.*

Camera Perturbations. Furthermore, the results in Table 3 demonstrate the advantage of using camera-base coordinate actions as prediction targets when there is variance in camera viewpoints within the fine-tuning data. Although the overall performance is slightly lower than that under strictly fixed-view conditions, the relative benefit of camera-based supervision increases, underscoring the generalizability of our approach in more realistic and variable settings.

Qualitative Comparison Figure 6 shows a comparison between OC-VLA and the baseline method under the robot base coordinate across different evaluation conditions and camera viewpoints. The results illustrate that OC-VLA offers improved robustness for fine-grained manipulation under a variety of settings. While baseline methods often fail to successfully complete tasks due to inaccurate grasp localization—especially under camera perturbations, OC-VLA consistently identifies more precise grasp positions. This advantage is particularly evident when there is variance in camera viewpoints: whereas baseline models begin to exhibit subtle errors, OC-VLA remains resilient and is able to complete the task successfully.

Conclusion

In this paper, we propose Observation-Centric VLA (OC-VLA), a simple yet effective framework that grounds action predictions in the camera base coordinate, addressing the spatial misalignment between perception and action in existing VLA models. OC-VLA introduces no architectural overhead and integrates seamlessly with existing pipelines. Extensive experiments show that OC-VLA significantly improves the cross-view generalization and enhance robustness under viewpoint shifts, showing the practical utility of OC-VLA and its strong potential for generalist robot policies.

Acknowledgments

This work is supported by the National Key R&D Program of China (NO.2022ZD0160201) and Young Scientists Fund (C Class) of China (NO.62503255). This work is supported by Shanghai Artificial Intelligence Laboratory.

References

Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; et al. 2024. pi_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Chen, X.; Choromanski, K.; Ding, T.; Driess, D.; Dubey, A.; Finn, C.; Florence, P.; Fu, C.; Arenas, M. G.; Gopalakrishnan, K.; Han, K.; Hausman, K.; Herzog, A.; Hsu, J.; Ichter, B.; Irpan, A.; Joshi, N.; Julian, R.; Kalashnikov, D.; Kuang, Y.; Leal, I.; Lee, L.; Lee, T.-W. E.; Levine, S.; Lu, Y.; Michalewski, H.; Mordatch, I.; Pertsch, K.; Rao, K.; Reymann, K.; Ryoo, M.; Salazar, G.; Sanketi, P.; Sermanet, P.; Singh, J.; Singh, A.; Soricut, R.; Tran, H.; Vanhoucke, V.; Vuong, Q.; Wahid, A.; Welker, S.; Wohlhart, P.; Wu, J.; Xia, F.; Xiao, T.; Xu, P.; Xu, S.; Yu, T.; and Zitkovich, B. 2023a. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. arXiv:2307.15818.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; Ibarz, J.; Ichter, B.; Irpan, A.; Jackson, T.; Jesmonth, S.; Joshi, N. J.; Julian, R.; Kalashnikov, D.; Kuang, Y.; Leal, I.; Lee, K.-H.; Levine, S.; Lu, Y.; Malla, U.; Manjunath, D.; Mordatch, I.; Nachum, O.; Parada, C.; Peralta, J.; Perez, E.; Pertsch, K.; Quiambao, J.; Rao, K.; Ryoo, M.; Salazar, G.; Sanketi, P.; Sayed, K.; Singh, J.; Sontakke, S.; Stone, A.; Tan, C.; Tran, H.; Vanhoucke, V.; Vega, S.; Vuong, Q.; Xia, F.; Xiao, T.; Xu, P.; Xu, S.; Yu, T.; and Zitkovich, B. 2023b. RT-1: Robotics Transformer for Real-World Control at Scale. arXiv:2212.06817.

Cheang, C.-L.; Chen, G.; Jing, Y.; Kong, T.; Li, H.; Li, Y.; Liu, Y.; Wu, H.; Xu, J.; Yang, Y.; et al. 2024. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*.

Chen, L.; Bahl, S.; and Pathak, D. 2023. Playfusion: Skill acquisition via diffusion from language-annotated play. In *Conference on Robot Learning*, 2012–2029. PMLR.

Dalal, M.; Chiruvolu, T.; Chaplot, D.; and Salakhutdinov, R. 2024. Plan-seq-learn: Language model guided rl for solving long horizon robotics tasks. *arXiv preprint arXiv:2405.01534*.

Dasari, S.; Mees, O.; Zhao, S.; Srirama, M. K.; and Levine, S. 2024. The ingredients for robotic diffusion transformers. *arXiv preprint arXiv:2410.10088*.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34: 8780–8794.

Gu, J.; Xiang, F.; Li, X.; Ling, Z.; Liu, X.; Mu, T.; Tang, Y.; Tao, S.; Wei, X.; Yao, Y.; et al. 2023. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*.

Ha, H.; Florence, P.; and Song, S. 2023. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, 3766–3777. PMLR.

Ho, J.; Jain, A.; and Abbeel, P. 2020a. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Ho, J.; Jain, A.; and Abbeel, P. 2020b. Denoising Diffusion Probabilistic Models. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Hou, Z.; Zhang, T.; Xiong, Y.; Duan, H.; Pu, H.; Tong, R.; Zhao, C.; Zhu, X.; Qiao, Y.; Dai, J.; and Chen, Y. 2025. Dita: Scaling Diffusion Transformer for Generalist Vision-Language-Action Policy. *arXiv preprint arXiv:2503.19757*.

Huang, S.; Chen, L.; Zhou, P.; Chen, S.; Jiang, Z.; Hu, Y.; Gao, P.; Li, H.; Yao, M.; and Ren, G. 2025. EnerVerse: Envisioning Embodied Future Space for Robotics Manipulation. *arXiv preprint arXiv:2501.01895*.

Jin, Y.; Li, D.; Shi, J.; Hao, P.; Sun, F.; Zhang, J.; Fang, B.; et al. 2024. Robotgpt: Robot manipulation learning from chatgpt. *IEEE Robotics and Automation Letters*, 9(3): 2543–2550.

Khazatsky, A.; Pertsch, K.; Nair, S.; Balakrishna, A.; Dasari, S.; Karamcheti, S.; Nasiriany, S.; Srirama, M. K.; Chen, L. Y.; Ellis, K.; et al. 2024. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.; Lam, G.; Sanketi, P.; et al. 2024. OpenVLA: An Open-Source Vision-Language-Action Model. *arXiv preprint arXiv:2406.09246*.

Kroemer, O.; Niekum, S.; and Konidaris, G. 2021. A review of robot learning for manipulation: Challenges, representations, and algorithms. *Journal of machine learning research*, 22(30): 1–82.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, 19730–19742. PMLR.

Li, P.; Wu, H.; Huang, Y.; Cheang, C.; Wang, L.; and Kong, T. 2025. GR-MG: Leveraging Partially-Annotated Data Via Multi-Modal Goal-Conditioned Policy. *IEEE Robotics and Automation Letters*.

Li, X.; Zhang, M.; Geng, Y.; Geng, H.; Long, Y.; Shen, Y.; Zhang, R.; Liu, J.; and Dong, H. 2024. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18061–18070.

Liu, S.; Wu, L.; Li, B.; Tan, H.; Chen, H.; Wang, Z.; Xu, K.; Su, H.; and Zhu, J. 2024. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Mu, T.; Ling, Z.; Xiang, F.; Yang, D.; Li, X.; Tao, S.; Huang, Z.; Jia, Z.; and Su, H. 2021. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. *arXiv preprint arXiv:2107.14483*.

Myers, V.; He, A. W.; Fang, K.; Walke, H. R.; Hansen-Estruch, P.; Cheng, C.-A.; Jalobeanu, M.; Kolobov, A.; Dragan, A.; and Levine, S. 2023. Goal representations for instruction following: A semi-supervised language interface to control. In *Conference on Robot Learning*, 3894–3908. PMLR.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

O'Neill, A.; Rehman, A.; Maddukuri, A.; Gupta, A.; Padalkar, A.; Lee, A.; Pooley, A.; Gupta, A.; Mandlekar, A.; Jain, A.; et al. 2024. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 6892–6903. IEEE.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4195–4205.

Perez, E.; Strub, F.; De Vries, H.; Dumoulin, V.; and Courville, A. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Reuss, M.; Li, M.; Jia, X.; and Lioutikov, R. 2023. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Shridhar, M.; Manuelli, L.; and Fox, D. 2022. Cliport: What and where pathways for robotic manipulation. In *Conference on robot learning*, 894–906. PMLR.

Shridhar, M.; Manuelli, L.; and Fox, D. 2023. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, 785–799. PMLR.

Singh, I.; Blukis, V.; Mousavian, A.; Goyal, A.; Xu, D.; Tremblay, J.; Fox, D.; Thomason, J.; and Garg, A. 2023. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 11523–11530. IEEE.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Team, O. M.; Ghosh, D.; Walke, H.; Pertsch, K.; Black, K.; Mees, O.; Dasari, S.; Hejna, J.; Kreiman, T.; Xu, C.; et al. 2024. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*.

Tian, Y.; Yang, S.; Zeng, J.; Wang, P.; Lin, D.; Dong, H.; and Pang, J. 2024. Predictive inverse dynamics models are scalable learners for robotic manipulation. *arXiv preprint arXiv:2412.15109*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Walke, H. R.; Black, K.; Zhao, T. Z.; Vuong, Q.; Zheng, C.; Hansen-Estruch, P.; He, A. W.; Myers, V.; Kim, M. J.; Du, M.; et al. 2023. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, 1723–1736. PMLR.

Wang, Z.; Li, Z.; Mandlekar, A.; Xu, Z.; Fan, J.; Narang, Y.; Fan, L.; Zhu, Y.; Balaji, Y.; Zhou, M.; et al. 2024. One-Step Diffusion Policy: Fast Visuomotor Policies via Diffusion Distillation. *arXiv preprint arXiv:2410.21257*.

Wen, J.; Zhu, M.; Zhu, Y.; Tang, Z.; Li, J.; Zhou, Z.; Li, C.; Liu, X.; Peng, Y.; Shen, C.; et al. 2024. Diffusion-VLA: Scaling Robot Foundation Models via Unified Diffusion and Autoregression. *arXiv preprint arXiv:2412.03293*.

Wen, J.; Zhu, Y.; Li, J.; Tang, Z.; Shen, C.; and Feng, F. 2025. DexVLA: Vision-Language Model with Plug-In Diffusion Expert for General Robot Control. *arXiv preprint arXiv:2502.05855*.

Wu, H.; Jing, Y.; Cheang, C.; Chen, G.; Xu, J.; Li, X.; Liu, M.; Li, H.; and Kong, T. 2023. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*.

Xia, F.; Li, C.; Martín-Martín, R.; Litany, O.; Toshev, A.; and Savarese, S. 2020. Relmogen: Leveraging motion generation in reinforcement learning for mobile manipulation. *arXiv preprint arXiv:2008.07792*.

Yamada, J.; Lee, Y.; Salhotra, G.; Pertsch, K.; Pflueger, M.; Sukhatme, G.; Lim, J.; and Englert, P. 2021. Motion planner augmented reinforcement learning for robot manipulation in obstructed environments. In *Conference on Robot Learning*, 589–603. PMLR.