Towards Efficient and Robust Manipulation via Multi-Frame Vision-Language-Action Modeling

Hao Li[1,2*], **Shuai Yang**[2,3*], **Yilun Chen**[2†], **Xinyi Chen**[2], **Xiaoda Yang**[3], **Yang Tian**[2], **Hanqing Wang**[2], **Tai Wang**[2], **Dahua Lin**[4], **Feng Zhao**[1†], **Jiangmiao Pang**[2†]

[1]University of Science and Technology of China
[2]Shanghai Artificial Intelligence Laboratory
[3]Zhejiang University
[4]The Chinese University of Hong Kong

Abstract

Recent vision-language-action (VLA) models built on pretrained vision-language models (VLMs) have demonstrated strong performance in robotic manipulation. However, these models remain constrained by the single-frame image paradigm and fail to fully leverage the temporal information offered by multi-frame histories, as directly feeding multiple frames into VLM backbones incurs substantial computational overhead and inference latency. We propose **CronusVLA**, a unified framework that extends single-frame VLA models to the multi-frame paradigm. CronusVLA follows a two-stage process: **(1) Single-frame pretraining** on large-scale embodied datasets with autoregressive prediction of action tokens, establishing an effective embodied vision-language foundation; **(2) Multi-frame post-training**, which adapts the prediction of the vision-language backbone from discrete tokens to learnable features, and aggregates historical information via feature chunking. CronusVLA effectively addresses the existing challenges of multi-frame modeling while enhancing performance. To evaluate the robustness under temporal and spatial disturbances, we introduce **SimplerEnv-OR**, a novel benchmark featuring 24 types of observational disturbances and 120 severity levels. Experiments across three embodiments in simulated and real-world environments demonstrate that CronusVLA achieves leading performance and superior robustness, with a 70.9% success rate on SimplerEnv, a 26.8% improvement over OpenVLA on LIBERO, and the highest robustness score on SimplerEnv-OR, showing the promise of efficient multi-frame adaptation for real-world VLA deployment.

Code — https://lihaohn.github.io/CronusVLA.github.io

1 Introduction

The rise of vision-language models (VLMs) (Karamcheti et al. 2024; Wang et al. 2024) has paved the way for general vision-language-action (VLA) models by offering powerful backbones and pretrained vision-language representations. Recent VLA methods (Kim et al. 2025; Qu et al. 2025; Zheng et al. 2025) primarily adapt advanced VLMs on large-scale heterogeneous manipulation datasets (O'Neill et al. 2024; Khazatsky et al. 2024) by re-engineering the tokenizer, while others (Black et al. 2024; Li et al. 2024a) draw inspiration from low-level policy designs, incorporating techniques like specialized heads and action chunks for better performance.

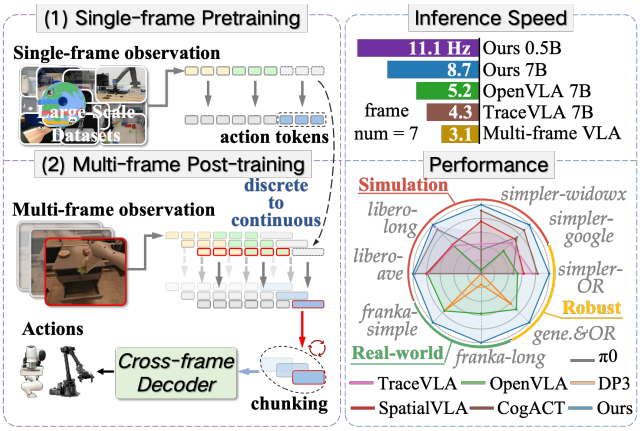

Figure 1: **CronusVLA** is a multi-frame modeling framework that includes single-frame pretraining on large-scale manipulation datasets and multi-frame post-training on cross-embodiment datasets. CronusVLA shows fast inference, high performance, and robustness in simulation and real world.

Low-level policies (Wu et al. 2024; Tian et al. 2024) have shown that leveraging multi-frame historical observations (i.e., RGB images captured from a single viewpoint over time) can considerably enhance performance. In fact, multi-frame information offers two notable advantages: (1) motion cues derived from consecutive observations *help determine the current execution phase* and effectively resolve state ambiguities. (2) actions can be *reliably inferred from prior consistent observations*, even when the current input is corrupted, showing strong observational robustness during execution.

However, most existing VLA models (Brohan et al. 2023a; Kim et al. 2025; Qu et al. 2025), built on the single-frame paradigm of vision-language models (VLMs), are typically trained using only a single current observation. Directly feeding multiple historical observation images into VLA mod-

*These authors contributed equally.
†Corresponding authors.

els introduces two major challenges: (1) the self-attention computation in VLM backbones scales quadratically with the number of input tokens, making large-scale embodied pretraining computationally expensive; and (2) redundant visual tokens considerably degrade inference speed, limiting the feasibility of real-world deployment. RoboVLMs (Li et al. 2024b) attempt to extend single-frame pretrained VLMs to the multi-frame paradigm by adopting memory-based LSTMs (Sak et al. 2014) and training embodiment capabilities from scratch, following standard policy learning paradigms. However, they overlook the potential benefits of efficiently adapting single-frame pretrained models to the multi-frame paradigm, and lack quantitative evaluation of the performance gains brought by such multi-frame capabilities.

To facilitate efficient multi-frame modeling, we propose **CronusVLA**, a unified framework for multi-frame training and inference, as illustrated in Fig.1. The approach comprises two key components: **(1) Single-frame Pretraining**: We first train a basic single-frame VLA model using standard autoregressive prediction over discrete action tokens, enabling more convenient utilization of large-scale heterogeneous embodied datasets and establishing an effective vision-language foundation. **(2) Multi-frame Post-training**: We introduce learnable features in the basic single-frame VLA model and conduct post-training on high-quality cross-embodiment datasets. This multi-frame post-training will replace multiple discrete action tokens with continuous learnable features and adapt the prediction of vision-language backbones from single-frame awareness to multi-frame. Extracting motion cues from multiple historical frames into a **feature chunking** enables effective temporal information aggregation, thereby improving efficiency. We further introduce the **feature modulator** and **multi-frame regularization**, which mitigate temporal imbalance and enhance convergence capability by reconstructing the influence of past frames within the model.

Current benchmarks, such as SimplerEnv (Li et al. 2025) and LIBERO (Liu et al. 2023), primarily evaluate VLA models across diverse tasks, objects, and scenes. They overlook the impact of observational disturbances on VLA models, which is a critical issue for future real-world applications. We introduce the **SimplerEnv-OR** (Observational Robustness) benchmark, which enables quantitative evaluation of model robustness beyond standard training data augmentation, considering both temporal and spatial disturbances. It includes 24 types of observational disturbances with over 120 levels of severity, and evaluates performance across 2,300 trials.

In total, our approach offers three advantages: (1) Fast inference. CronusVLA predicts learnable features in a single forward pass without relying on autoregressive prediction, enabling faster action generation over previous VLA models. Cached historical learnable features further eliminate redundant computation during inference, resulting in substantial speed improvements. (2) High performance. Extensive experiments across three embodiments and diverse manipulation tasks in both the simulation and real world are conducted. CronusVLA achieves state-of-the-art performance on the simulation benchmark SimplerEnv with an average 70.9% success rate, a 26.8% overall improvement over OpenVLA on LIBERO. (3) Strong robustness. CronusVLA can maintain strong robustness against temporal and spatial disturbances. In our real-world experiments, it achieves a 72.6% success rate under various challenges involving interference and occlusion. Our model also surpasses previous approaches on both robustness score and success rate of SimplerEnv-OR.

Our contributions are summarized as follows:

- We propose a unified framework, CronusVLA, to extend VLA models to a multi-frame paradigm, which includes single-frame pertaining and multi-frame post-training.
- We propose SimplerEnv-OR, a novel benchmark that enables quantitative evaluation of VLA models' robustness under observational disturbances.
- Extensive experiments have demonstrated CronusVLA's leading performance and strong observational robustness across simulation and the real world.

2 Related Works

Vision-Language-Action models. Current VLA models usually integrate action generation into pretrained VLMs (Karamcheti et al. 2024; Abdin et al. 2024). Via an action tokenizer, RT-2 (Brohan et al. 2023a) and OpenVLA discretize 7D actions and generate them via autoregressive prediction. SpatialVLA (Qu et al. 2025) unifies the action space of various robots via adaptive action grids. 3D-VLA (Zhen et al. 2024) and Magma (Yang et al. 2025a) unify action prediction and multimodal tasks within a single model. They minimally adapt VLMs into a general manipulation policy by treating actions as discrete tokens. Instead, the remaining VLAs (Wen et al. 2025; Yang et al. 2025b) abandon the original discrete paradigm of VLMs. They augment VLMs with additional action heads (Sak et al. 2014; Peebles and Xie 2023), and train embodiment capabilities and continuous action prediction from scratch. Our method focuses on transferring a single-frame, discretely pretrained VLA model to the multi-frame paradigm with continuous action prediction.
Multi-frame modeling for manipulation. Most VLAs (Kim et al. 2025; Driess et al. 2023; Brohan et al. 2023a) treat each action prediction as a temporally independent decision, and are also trained in a single-frame manner. While low-level policies (Chi et al. 2025; Zhao et al. 2023) incorporate multi-frame modeling by processing multiple images simultaneously based on their lightweight architectures. Other policies (Cheang et al. 2024; Tian et al. 2024) are pretrained on large-scale video datasets (Miech et al. 2019; Grauman et al. 2022) by interleaving multi-step action and multi-frame image prediction. However, directly applying this strategy to large-scale VLAs introduces considerable computational overhead. To address these limitations, Dita (Hou et al. 2025) employs multiple RGB images in a small backbone. TraceVLA (Zheng et al. 2025) uses visual prompting to draw past traces on the current observation. RoboFlamingo (Li et al. 2024c) and RoboVLMs (Li et al. 2024b) primarily adopt memory-based LSTMs and training embodiment capabilities from scratch to model temporal relations across frames. In contrast, our method builds on a single-frame pretrained VLA model and explicitly establishes multi-frame capabilities during post-training, which retains the single-frame perception meanwhile enabling multi-frame modeling.

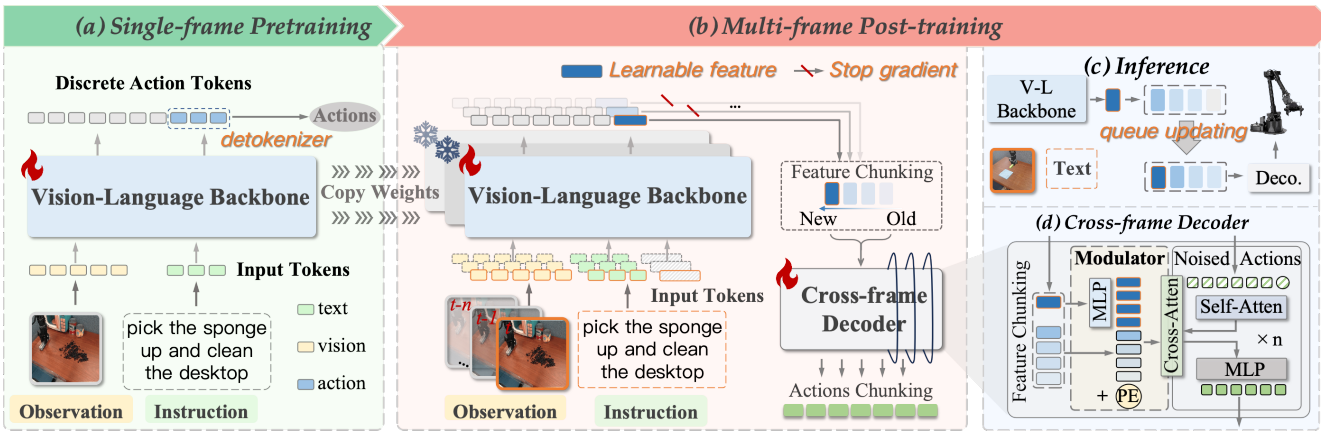

Figure 2: **Overview of CronusVLA framework.** (a) illustrates the single-frame pretraining of the basic single-frame VLA. By duplicating the model weights, we perform multi-frame post-training as shown in (b), where multi-frame modeling is achieved by aggregating learnable features from several preceding frames in a cross-frame decoder. In (c), a queue mechanism is conducted on feature chunking for fast inference. Details of the cross-frame decoder are illustrated in (d).

3 Methodology

We describe the single-frame training process in Sec.3.1. In Sec.3.2, we detail the multi-frame post-training, including the vision-language backbone, the cross-frame decoder, and multi-frame regularization. Finally, Sec.3.3 shows our SimplerEnv-OR benchmark. The overview is shown in Fig.2.

3.1 Single-frame Pretraining

As shown in Fig.2 (a), our first step is to establish a basic single-frame VLA model. Off-the-shelf pretrained VLMs (Karamcheti et al. 2024; Wang et al. 2024) are adapted by learning large-scale manipulation demonstrations $D_i = \{I_t, a_t, l\})_{t=0}^{T_i}$, where T_i is the length of episode i, l is the language instruction, observation I_t denotes a RGB image from a single camera at step t, and $a_t \in \mathbb{R}^n$ represents the corresponding actions. Following (Kim et al. 2025), discrete action tokens are derived from continuous actions a_t via the extended action tokenizer, which maps actions into 256 bins, and are trained using the next-token prediction objective. Given I_t and l, the model predicts the next-step action tokens and detokenizes them, $a_t = \text{VLA}(I_t, l)$. We observe that the single-frame pretraining effectively transfers the visual perception capabilities of vision encoders SigLip (Zhai et al. 2023) and Dinov2 (Oquab et al. 2023) to embodied scenes, which provides an effective vision-language foundation for multi-frame post-training. Meanwhile, single-frame pretraining can better preserve single-frame visual perception of VLMs and incur lower training costs on large-scale data, compared to directly pretraining in a multi-frame manner.

3.2 Multi-frame Post-traing

From discrete action tokens to feature chunking. For our basic single-frame VLA, vision tokens $\{v^i, i \in [0, n_v]\}$ and text tokens $\{l^i, i \in [0, n_l]\}$ are causally computed in the vision-language backbone, which autoregressively predict discrete action tokens by summarizing information from all previous tokens. As shown in Fig.2 (b), during multi-frame post-training, instead of generating discrete action tokens a_t, we introduce learnable features $f_t \in \mathbb{R}^d$ in the backbone's hidden layers as continuous representations. This feature is designed to integrate the pretrained model's embodied vision-language summarization capability and is computed as $f_t = \text{VL}(I_t, l)$. All images are still encoded by our vision-language backbone within a single-frame formulation, ensuring compatibility with standard VLM formulations. We introduce *feature chunking* $F_t^M = \{f_{t-M+1}, \ldots, f_{t-1}, f_t\} = f_{t-M+1:t}$ to establish the association between different frames. It is a chunking of historical learnable features and can represent multi-frame images of M steps at the feature-level. During training, we restructure M-step image inputs at the batch level, enabling the vision-language backbone to independently process $B \times M$ single-frame inputs per iteration, where B denotes the original batch size. During inference, as shown in Fig.2 (c), we maintain the feature chunking using a first-in, first-out queue mechanism, which ensures fast inference by reusing prior vision-language computations.

Cross-frame decoder. The cross-frame decoder predicts action chunking by decoding the multi-frame information embedded in the feature chunking F_t^M, $a_{t:t+K-1} = \text{Decoder}(F_t^M)$, as shown in Fig.2 (d). We construct a DiT-based decoder composed of self-attention networks and MLP layers, and train it using a diffusion loss $\mathcal{L}$. To balance the contributions of the current and past learnable features, we employ a feature *modulator* to dynamically modulate the learnable features. Specifically, the current feature $f_t \in \mathbb{R}^d$ is divided to match the number of past learnable features $f_{t-M+1:t-1}$ through channels splitting (DIV), and then processed by modulator (MD) to get the modulated feature Z_f:

$$Z_f = \text{MD}(F_t^M) = \text{MLP}\left(f_{t-M+1:t-1}, \tilde{f}_t\right) \quad (1)$$

$$\tilde{f}_t = \text{DIV}(f_t), \text{where } f_t \in \mathbb{R}^d, \tilde{f}_t \in \mathbb{R}^{(M-1)\times d}, \quad (2)$$

where DIV consists of a dimensionality-expanding Linear layer followed by a feature-splitting operation, $Z_f \in$

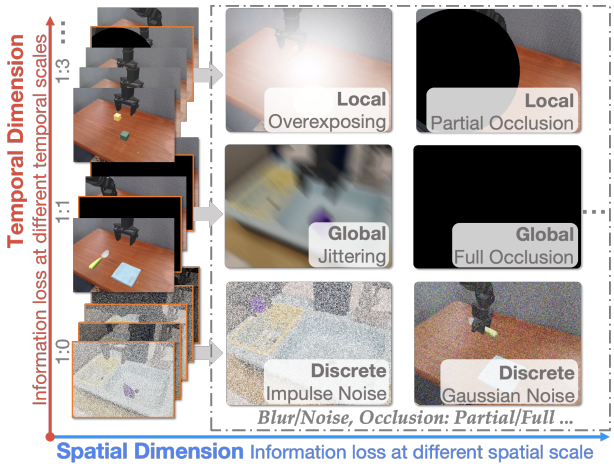

Figure 3: An illustration of the SimplerEnv-OR benchmark.

$\mathbb{R}^{2\cdot(M-1)\times d'}$, d' is the hidden dimension of the decoder. We further adopt a cross-attention mechanism to process noised actions and modulated features, which enables effective interaction. Specifically, Z_f is fed into the cross-attention network and mapped to the keys and values, where noised actions $\hat{a}$ serve as queries. Noised actions are iteratively denoised conditioned on Z_f for the final action output during inference.

Post-training with multi-frame regularization. We introduce *the multi-frame regularization* to decouple the vision-language backbone from multi-frame modeling of the decoder, ensuring its training logic remains consistent with the single-frame paradigm. Specifically, the past learnable features $f_{t-M+1:t-1}$ in the feature chunking F_t^M are treated as auxiliary inputs to the decoder, with their influence limited within the decoder, where t is the execution step. Their gradient flows do not update the vision-language backbone, and serves solely as a regularization term to facilitate training:

$$\hat{f}_{t-M+1:t-1} = \left\{\mathrm{sg}\big(\mathrm{VL}(I_{t-k}, l)\big), k = 1, \ldots, M-1\right\}, \quad (3)$$

$$\mathcal{L} = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbf{I}),i}\left[\left\|\hat{\epsilon}^i - \epsilon_\theta\left(t,\ \hat{f}_{t-M+1:t-1},\ f_t\right)\right\|_2\right], \quad (4)$$

where sg means the stop-gradient operation, VL is the vision-language backbone, $\mathcal{L}$ is the diffusion loss (multi-step denoising), $\hat{\epsilon}^i$ is the predicted i-step noise, associated with the noised actions $\hat{a}^i_{t:t+K-1}$. This method offers two advantages: (1) Extracting past learnable features without gradient computation reduces computational and memory overhead, enabling efficient training. (2) Updating on a single-frame basis preserves the pretrained single-frame perceptual capabilities and promotes faster convergence.

3.3 SimplerEnv-OR Benchmark

As shown in Fig.3, we introduce the SimplerEnv-OR benchmark. It is designed for quantitatively evaluating the observational robustness of VLAs by simulating different observational disturbances of the camera, which is beyond the simple visual augmentation strategies used in training.

Evaluation settings. SimplerEnv-OR extends the simulation environment of the WidowX Robot Visual Matching (WR-VM) setting in SimplerEnv (Li et al. 2025), and evaluates models trained on Bridge-v2 (Walke et al. 2023). We consider two disturbance dimensions: spatial and temporal. The spatial dimension introduces visual disturbances in different positions to simulate real-world camera artifacts, including Global (i.e., blurring, jittering, full occlusion), Local (i.e., overexposing, partial occlusions), and Discrete (i.e., noise, impulse). The temporal dimension assesses robustness under varying disturbance frequencies, including Constant (1:0), Cyclic (1:1), and Sparse (1:3, 1:5). Under all disturbance settings, VLAs must complete the same tasks as WR-VM.

Robustness score. To quantify robustness, we define a robustness score $R\text{-}Score$ based on relative performance maintenance under various disturbances. Let SR denote the average success rate of the original WR-VM tasks, and SR^i denote the success rate under the disturbance setting i. The robustness score is computed as: $R\text{-}Score^i = 100 * \frac{SR^i}{SR}$. Noting that each setting includes 200 to 400 trials, ensuring a stable and reliable evaluation.

4 Experiment

4.1 Results in Simulation

Implementation details. In this section, we primarily investigate the performance of our post-trained model. After pretraining the basic single-frame VLA following (Kim et al. 2025; Belkhale and Sadigh 2024) with the OXE dataset (O'Neill et al. 2024), we select two high-quality datasets, Bridge-v2 (Walke et al. 2023) and Fractal (Brohan et al. 2023b) datasets, to conduct multi-frame post-training, which include about 148k episodes and 5M multi-frame clips. Our CronusVLA 7B is built on 7B Llama 2 (Touvron et al. 2023), and CronusVLA 0.5B is built on Qwen2.5 0.5B (Yang et al. 2024), they both employ Dinov2 and SigLip as vision encoders. CronusVLA is built on a third-person camera and a text instruction. CronusVLA 7B is configured with a default of 6 past frames, while 0.5B utilizes 3 past frames.

Evaluation setup in SimplerEnv. We conduct experiments within SimplerEnv, a benchmark to evaluate models with the WidowX Robot (WR) and the Google Robot (GR). SimplerEnv covers more than 2k trails across different scenarios, objects, and tasks. We report the average success rate of each task. RT-1-X (O'Neill et al. 2024; Brohan et al. 2023b), RT-2-X (O'Neill et al. 2024; Brohan et al. 2023a), and Octo-Based (Octo Model Team et al. 2024) are early baselines, OpenVLA, CogACT, OpenVLA-OFT (Kim, Finn, and Liang 2025) and Magma (Yang et al. 2025a) are trained on subsets of the OXE. π_0 (Black et al. 2024), π_0-FAST (Pertsch et al. 2025) and GR00T-N1.5 (Bjorck et al. 2025) are VLAs with additional robot state input. RoboVLMs, SpatialVLA, and TraceVLA are trained on the Fractal and Bridge-V2 datasets.

Experimental results on SimplerEnv. Main results are illustrated in Tab.1. For Google Robot setting, our CronusVLA-7B achieves the highest average success rate, with 78.6 in the VM and 73.8 in the VA, surpassing TraceVLA and RoboVLMs (also multi-frame VLAs), by +71.6% and +37.9% relative VM score, +48.2% and +138.8% relative VA scores. Notably, our model achieves strong performance on not only simple tasks *Pick Coke Can* and *Open/Close Drawer*, but also the long-horizon task *Put in Drawer*, which requires

Methods	Google Robot										WidowX Robot					
	Open/Close Drawer		Put in Drawer		Pick Coke Can		Move Near		**Avg**		Put Spoon	Put Carrot	Stack Blocks	Put Eggplant	**Avg**	**Avg**
	VM	VA	VM	VA	VM	VA	VM	VA	VM	VA			VM			
RT-1-X*	59.7	29.4	21.3	10.1	56.7	49.0	31.7	32.3	42.4	30.2	0.0	4.2	0.0	0.0	1.1	24.6
RT-2-X*	25.0	35.5	3.7	20.6	78.7	82.3	77.9	79.2	46.3	54.4	-	-	-	-	-	-
Octo-Base*	22.7	1.1	0.0	0.0	17.0	0.6	4.2	3.1	11.0	1.2	15.8	12.5	0.0	41.7	17.5	9.9
RoboVLMs (2B)	44.9	10.3	27.8	0.0	76.3	50.7	79.0	62.5	57.0	30.9	50	37.5	0.0	83.3	42.7	43.5
SpatialVLA (3B)*	**54.6**	**39.2**	0.0	6.3	79.3	78.7	90.0	**83.0**	56.0	51.8	20.8	37.5	41.7	**83.3**	45.8	51.2
π_0 (3B)	38.3	25.6	-	-	72.7	75.2	65.3	63.7	-	-	25.0	16.7	12.5	29.2	20.9	-
π_0-FAST (3B)	42.9	31.3	-	-	75.3	77.6	67.5	68.2	-	-	29.1	21.9	10.8	66.6	32.1	-
GR00T-N1.5 (2B)	27.8	35.8	7.4	4.0	51.7	69.3	54.0	68.7	35.2	44.5	**75.3**	**54.3**	**57.0**	61.3	**61.9**	47.2
Ours (0.5B)*	50.5	36.8	**42.6**	**21.7**	**96.0**	**94.6**	**93.0**	78.0	**70.5**	**57.8**	45.8	33.3	0.0	79.2	39.6	**56.0**
OpenVLA (7B)*	59.7	23.5	0.0	2.9	25.7	54.1	55.0	63.0	35.1	35.9	8.3	4.2	0.0	0.0	3.1	24.7
CogACT (7B)*	71.8	28.3	50.9	46.6	91.3	89.6	**85.0**	**80.8**	74.8	61.3	**75.0**	50.0	16.7	79.2	55.2	63.8
TraceVLA (7B)*	63.1	**61.6**	11.1	12.5	45.0	64.3	63.8	60.6	45.8	49.8	12.5	16.6	16.6	65.0	27.7	41.1
OpenVLA-OFT (7B)	47.2	12.2	0.9	0.5	72.3	65.3	69.6	59.0	47.5	34.3	12.5	4.2	4.2	100.0	30.2	37.3
Magma (8B)*	58.9	59.0	8.3	24.0	75.0	68.6	53.0	78.5	48.8	57.5	37.5	29.2	20.8	91.7	44.8	50.4
Ours (7B)*	**77.8**	58.7	**64.8**	**65.1**	**95.7**	**94.2**	76.0	77.0	**78.6**	**73.8**	66.7	**54.2**	**20.8**	**100.0**	**60.4**	**70.9**

Table 1: **Performance comparison on SimplerEnv.** The Google Robot environment includes two settings, Visual Matching (VM) and Variant Aggregation (VA). WidowX Robot environment only includes the VM settings. Experiments are conducted across 12 tasks. Sizes of the original LLMs are listed, with % omitted. * indicates evaluating all 12 tasks using a co-trained checkpoint; otherwise, two checkpoints are used to evaluate the Google Robot and WidowX setups separately.

Methods	Spatial	Object	Goal	Long	Ave.
Dita	84.2	96.3	85.4	63.8	82.4
OpenVLA	84.7	88.4	79.2	53.7	76.5
TraceVLA	84.6	85.2	75.1	54.1	74.8
SpatialVLA	88.2	89.9	78.6	55.5	78.1
π_0-FAST	96.4	96.8	88.6	60.2	85.5
GR00T-N1	94.4	97.6	93.0	90.6	93.9
π_0	96.8	98.8	95.8	85.2	94.2
$\pi_{0.5} + KI$	**98.0**	97.8	95.6	85.8	94.3
Ours(7B)	97.3	**99.6**	**96.9**	**94.0**	**97.0**

Table 2: **Results on LIBERO**, average success rates reported.

	Temporal Dimension						*WidowX Robot*	
	Constant (1:0)		Cyclic (1:1)		Sparse (1:3)		*VM*	
Methods	R-Score	SR	R-Score	SR	R-Score	SR	-	*SR*
π_0	43.5	9.1	36.8	7.7	34.9	7.3	-	*20.9*
TraceVLA	59.2	16.4	62.5	17.3	78.0	21.6	-	*27.7*
RoboVLMs	47.6	20.3	57.3	24.5	78.7	33.6	-	*42.7*
SpatialVLA	44.9	20.6	48.0	22.0	63.1	28.9	-	*45.8*
CogACT	53.3	29.4	66.1	36.5	80.2	44.3	-	*55.2*
Ours (7B)	**61.2**	**37.0**	**86.7**	**52.3**	**96.2**	**58.1**	-	*60.4*

	Spatial Dimension						Total Avg.	
	Global		Local		Discrete			
Methods	R-Score	SR	R-Score	SR	R-Score	SR	**R-Score**	SR
π_0	42.6	8.9	28.2	5.9	52.1	10.9	41.1	8.6
TraceVLA	58.1	16.1	65.3	18.1	81.9	22.7	65.8	18.2
RoboVLMs	54.7	23.4	83.3	35.6	76.8	32.8	67.4	28.8
SpatialVLA	57.6	26.4	50.0	22.9	52.4	24.0	54.4	24.9
CogACT	60.2	33.2	80.5	44.4	**87.4**	48.3	72.1	39.8
Ours (7B)	**85.4**	**51.6**	**96.6**	**58.3**	80.2	**48.4**	**86.9**	**52.4**

Table 3: **Observational robustness test on SimplerEnv-OR.** Top: Temporal dimension with different frequencies. Bottom: Spatial dimension with different patterns. R-Score is the predefined score of robustness. SR denotes the success rate (%).

sequential actions of opening the drawer and placing the apple in it. Most previous approaches have failed to attain high success rates in *Put in Drawer*, while our method effectively improves its VM success rate to 64.8 and the VA to 65.1. For the WidowX, CronusVLA 7B also achieves a high average success rate, +41.5% higher than SpatialVLA and +9.4% higher than CogACT. CronusVLA 0.5B, with a smaller language backbone, outperforms many prior models trained on larger (from 2B to 7B) language models. It achieves the best results in *Pick Coke Can* and *Move Near* over all other models, suggesting that excessive parameters may not always be beneficial, emphasizing the value of effective modeling.

Evaluation setup and experimental results in LIBERO. We evaluate CronusVLA on the LIBERO (Liu et al. 2023) benchmark. LIBERO comprises four task suites, including Spatial, Object, Goal, and Long. Based on the post-trained weights, CronusVLA is fully finetuned on each suite. We compare it with Dita (Hou et al. 2025), OpenVLA, TraceVLA, and SpatialVLA, as well as other models conditioned on an additional wrist-view image and the robot state, including π_0, π_0-FAST, GR00T-N1, and $\pi_{0.5}$ (Black et al. 2025). In Tab.2, our model with only an extra wrist-view input achieves the highest average success rate of 97.0%, attaining good performance across almost all suites and a remarkable 94.0% on Long (+40.3% over OpenVLA), which shows our robust and long-horizon learning capabilities.

Robustness testing on SimplerEnv-OR. We evaluate the observational robustness of several models: RoboVLMs (also a multi-frame VLA), SpatialVLA (spatial modeling VLA), CogACT (prior SOTA model), TraceVLA (designed for temporal modeling), π_0 (popular baseline). All results are summarized in Tab.3. In the *Temporal Dimension,* as the distur-

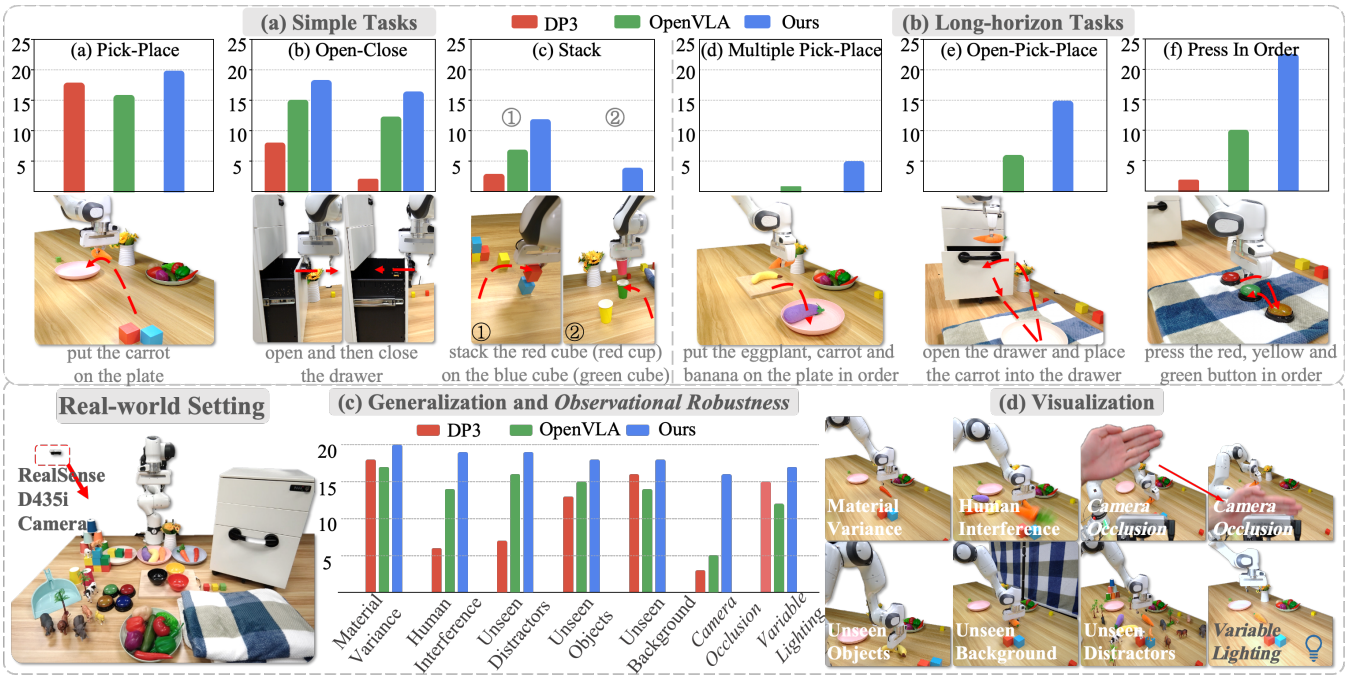

Figure 4: **Real-world experiment.** Evaluation of basic pick-and-place capabilities is shown in (a), long-horizon tasks of (b) demonstrate the advantages of multi-frame modeling in handling temporally dependent manipulations, and generalization and robustness tests of (c), particularly under camera occlusion and various disturbances, highlight the robustness of our model.

bance frequency decreases from Constant (1:0) to Sparse (1:3), all models exhibit improved R-Scores. Single-frame models (π_0, SpatialVLA and CogACT) tend to produce out-of-distribution actions under high-frequency disturbances, leading to task failure. RoboVLMs and TraceVLA, despite being multi-frame, heavily rely on accurate historical information and often default to inaction or repeated probing when disturbed. In contrast, CronusVLA demonstrates superior robustness due to effective multi-frame modeling, achieving strong resistance under Constant (1:0) and near immunity under Sparse (1:3) with 96.2 R-Score. In the *Spatial Dimension*, SpatialVLA is highly sensitive to local disturbances due to its reliance on zero-shot depth estimation, while CogACT and RoboVLMs are more affected by global disturbances. CronusVLA consistently outperforms others under these two types, exhibiting strong robustness. Notably, although SpatialVLA outperforms RoboVLMs on the original SimplerEnv benchmark, it underperforms on SimplerEnv-OR, with lower SR and R-Score, which highlights RoboVLMs' robustness advantage as a multi-frame model. Overall, SimplerEnv-OR provides quantitative evidence of CronusVLA's and other VLAs' robustness under observational disturbances.

4.2 Real-world Experimental Results

Evaluation setup within Franka platform. As shown in Fig.4, we evaluate CronusVLA on several real-world tasks with the Franka Research 3 Robot. We utilize a third-person camera for visual input. Three task suites are designed: *(1) Simple Tasks*, involves the picking and placing objects, opening and closing the drawer, and stacking cubes or cups; *(2) Long-horizon Tasks*, requires coordinated multi-step manipulation and includes putting multiple objects, placing objects into a drawer and pressing buttons in a specific order; and *(3) Generalization and Observational Robustness*, evaluating performance on unseen objects, camera occlusions, distractor objects and so on. We collect 50 episodes for each task, and success rates of 25 trials are reported. We train DP3 (Ze et al. 2024) and OpenVLA on these demonstrations, our CronusVLA 7B is finetuned from the post-trained weight.

Experimental results within Franka platform. CronusVLA outperforms other models across almost all tasks. For *Simple Tasks*, all models perform well when handling simple objects; however, for tasks requiring more precise manipulation, such as stacking blocks (or cups), CronusVLA shows better in-domain performance. For *Long-horizon Tasks*, CronusVLA exhibits stronger long-horizon learning capabilities, achieving consistently better performance than DP3 and OpenVLA under limited expert demonstrations. Notably, in *press buttons in order*, OpenVLA tends to press the same button multiple times, indicating state confusion in long-horizon tasks due to the absence of multi-frame information, while CronusVLA has inherent temporal awareness for button pressing. *Generalization and Observational Robustness* tasks are illustrated in Fig.4 (c) and (d). In all distracted situations, our model shows the best performance. DP3 is sensitive to distractions and human interference. In the *Camera Occlusions*, OpenVLA is adversely affected by frequent visual dropouts due to their reliance on precise per-step observations, while our multi-frame modeling effectively withstands such disturbances.

Settings	G-VM	G-VA	W-VM	Ave.	Speed (Hz)
Baseline	43.3	39.4	10.4	31.0	5.18
+M.F.	45.5	47.4	4.2	32.4	3.09
+M.F. +Dec.	52.2	58.1	34.4	48.2	8.73
+M.F. +Dec. +V.L.	72.7	72.6	56.3	67.2	8.73
+M.F. +Dec. +V.L. +Reg.	**78.6**	**73.8**	**60.4**	**70.9**	**8.73**

Table 4: **Ablation on post-training strategies.** Success rates on SimplerEnv and inference speeds are shown. We discuss adding multiple images, training decoder or vision-language backbone, and adding multi-frame regularization.

#	Settings	G-VM	G-VA	W-VM	Ave.
1	w/o. cross-atten	76.4	71.2	57.3	68.3
2	w/o. modulator	68.7	61.4	60.4	63.5
3	MLP-based decoder	66.4	53.6	35.4	51.8
4	SiT-based decoder	75.0	69.5	58.4	67.6
5	**Ours**	**78.6**	**73.8**	**60.4**	**70.9**

Table 5: **Ablation on cross-frame decoder designs.** Based on CronusVLA 7B with a past frame number of 6.

4.3 Ablations Study

Ablations on post-training strategies. As shown in Tab. 4, we evaluate the effectiveness of different multi-frame post-training strategies (with a total of 7 frames) by comparing the following configurations: (1) *Baseline*: Our basic single-frame VLA model, further post-trained without multi-frame modeling. (2) *+M.F.*: Directly feeding multiple images into the vision-language backbone and post-training in a discrete action space. This yields a margin performance gain (only +4.5%) but considerably reduces inference speed (-40.3%). (3) *+M.F. +Dec.*: Replacing discrete action prediction with continuous feature prediction, training only the decoder during multi-frame modeling. This setting achieved substantial performance improvement and considerably boosts inference speed (8.73 vs. 3.09 Hz), thanks to the elimination of autoregressive decoding and the caching of feature chunking. (4) *+M.F. +Dec. +V.L.*: Enabling training of both the decoder and vision-language backbone under multi-frame post-training also improves the overall performance, attributed to the enhanced fitting ability. (5) *+M.F. +Dec. +V.L. +Reg. (Ours)*: Utilizing the multi-frame regularization can effectively leverage single-frame perception of vision-language backbone while ensuring multi-frame modeling of the decoder, improving performance. We also observe that multi-frame regularization considerably speeds up the convergence.

Ablations on cross-frame decoder designs. As illustrated in Tab.5, we first study the core design in our DiT-based decoder. For *w/o. cross-atten*, we remove the cross-attention mechanism of our original design and replace it with a direct self-attention network. Our cross-attention design achieves superior overall performance and higher per-task success rates. Notably, self-attention incurs quadratic computational growth with increasing frame numbers, whereas our cross-attention mechanism scales linearly, enabling support for extremely large frame sequences. For *w/o. modulator*, we omit the modulator, treating current (1 frame) and past learnable features (6 frames) indiscriminately without feature splitting and learnable positional embedding, which substantially degrades performance due to the potential dominance of irrelevant historical information over critical current cues. Then, we explore alternative decoder architectures by adopting an *MLP-based decoder* with the same parameter size as our DiT-based model. Due to the large-scale post-training data, the limited representational capability of the MLP leads to a 20% drop in success rate. The *SiT-based decoder* (Ma et al. 2024) replaces the diffusion objective with a flow-matching one, yielding a lower average score than the DiT-based decoder.

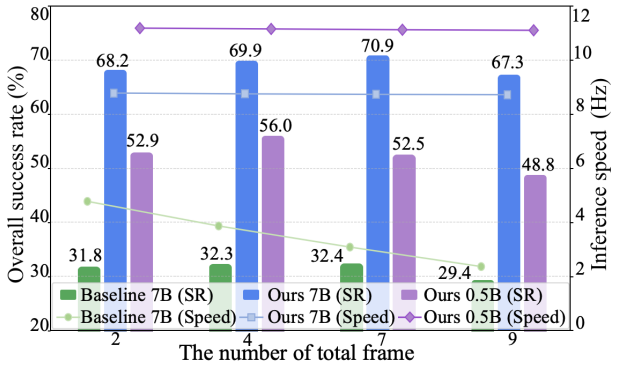

Figure 5: **Impact of frame number.** Varying the number of input frames affects the success rate and inference speed.

The impact of frame number. As shown in Fig.5, we analyze how varying the number of input frames affects different multi-frame modeling strategies. Given that tasks differ in their reliance on temporal information, we conduct a representative analysis on the SimplerEnv benchmark. The study compares CronusVLA 7B, CronusVLA 0.5B, and the Baseline model (post-trained basic single-frame VLA mode) with a naive multi-frame extension. Results show several conclusions: (1) Increasing the total frame number yields an initial increase followed by a subsequent decrease in average success rate. More frames do not consistently lead to better outcomes: CronusVLA 7B performs best with 7 frames, while the CronusVLA 0.5B performs best with 4, suggesting that a moderate amount of temporal information can enhance performance, whereas excessive temporal input may lead to performance degradation. (2) Compared with Baseline, CronusVLA maintains high inference speed across frame counts, avoiding excessive latency overhead. (3) CronusVLA demonstrates strong long-horizon efficiency, with inference speed of baseline models significantly degrading as the horizon extends, while our model remains largely unaffected.

5 Conclusion

We presented CronusVLA, a unified framework that extends single-frame VLA models to the multi-frame paradigm. By introducing feature chunking and multi-frame regularization, CronusVLA achieves strong performance and enhanced robustness. We also proposed SimplerEnv-OR, a benchmark for testing observational robustness. Experiments on simulation and the real world show the effectiveness of our method.

Acknowledgments

This work is funded in part by the National Key R&D Program of China (2022ZD0160201), Shanghai Artificial Intelligence Laboratory, the Special Program of the Graduate School, University of Science and Technology of China, and Anhui Provincial Natural Science Foundation under Grant 2108085UD12. We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC. The AI-driven experiments, simulations and model training were partly performed on the robotic AI-Scientist platform of Chinese Academy of Sciences.

References

Abdin, M.; Aneja, J.; Awadalla, H.; Awadallah, A.; Awan, A. A.; Bach, N.; Bahree, A.; Bakhtiari, A.; Bao, J.; Behl, H.; et al. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. Technical Report MSR-TR-2024-12, Microsoft.

Belkhale, S.; and Sadigh, D. 2024. MiniVLA: A Better VLA with a Smaller Footprint.

Bjorck, J.; Castañeda, F.; Cherniadev, N.; Da, X.; Ding, R.; Fan, L.; Fang, Y.; Fox, D.; Hu, F.; Huang, S.; et al. 2025. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots. *arXiv preprint arXiv:2503.14734*.

Black, K.; Brown, N.; Darpinian, J.; Dhabalia, K.; Driess, D.; Esmail, A.; Equi, M. R.; Finn, C.; Fusai, N.; et al. 2025. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. In *Proceedings of The 9th Conference on Robot Learning*, volume 305, 17–40.

Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; Jakubczak, S.; Jones, T.; Ke, L.; Levine, S.; Li-Bell, A.; Mothukuri, M.; Nair, S.; Pertsch, K.; Shi, L. X.; Tanner, J.; Vuong, Q.; Walling, A.; Wang, H.; and Zhilinsky, U. 2024. π_0: A Vision-Language-Action Flow Model for General Robot Control. arXiv:2410.24164.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Chen, X.; Choromanski, K.; Ding, T.; Driess, D.; Dubey, A.; Finn, C.; et al. 2023a. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of The 7th Conference on Robot Learning*, volume 229, 2165–2183.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; et al. 2023b. RT-1: Robotics Transformer for Real-World Control at Scale. In *Proceedings of Robotics: Science and Systems*.

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*.

Chi, C.; Xu, Z.; Feng, S.; Cousineau, E.; Du, Y.; Burchfiel, B.; Tedrake, R.; and Song, S. 2025. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44: 1684–1704.

Driess, D.; Xia, F.; Sajjadi, M. S.; Lynch, C.; Chowdhery, A.; Ichter, B.; Wahid, A.; Tompson, J.; Vuong, Q.; Yu, T.; et al. 2023. PaLM-E: an embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, 8469–8488.

Grauman, K.; Westbury, A.; Byrne, E.; Chavis, Z.; Furnari, A.; Girdhar, R.; Hamburger, J.; Jiang, H.; Liu, M.; Liu, X.; et al. 2022. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18995–19012.

Hou, Z.; Zhang, T.; Xiong, Y.; Duan, H.; Pu, H.; Tong, R.; Zhao, C.; Zhu, X.; Qiao, Y.; Dai, J.; et al. 2025. Dita: Scaling Diffusion Transformer for Generalist Vision-Language-Action Policy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7686–7697.

Karamcheti, S.; Nair, S.; Balakrishna, A.; Liang, P.; Kollar, T.; and Sadigh, D. 2024. Prismatic VLMs: investigating the design space of visually-conditioned language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, 23123–23144.

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. In *Proceedings of Robotics: Science and Systems*.

Kim, M. J.; Finn, C.; and Liang, P. 2025. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In *Proceedings of Robotics: Science and Systems*.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.; Lam, G.; Sanketi, P.; et al. 2025. OpenVLA: An Open-Source Vision-Language-Action Model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270, 2679–2713.

Li, Q.; Liang, Y.; Wang, Z.; Luo, L.; Chen, X.; Liao, M.; Wei, F.; Deng, Y.; Xu, S.; Zhang, Y.; et al. 2024a. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*.

Li, X.; Hsu, K.; Gu, J.; Pertsch, K.; Mees, O.; Walke, H. R.; Fu, C.; Lunawat, I.; Sieh, I.; Kirmani, S.; et al. 2025. Evaluating Real-World Robot Manipulation Policies in Simulation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270, 3705–3728.

Li, X.; Li, P.; Liu, M.; Wang, D.; Liu, J.; Kang, B.; Ma, X.; Kong, T.; Zhang, H.; and Liu, H. 2024b. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*.

Li, X.; Liu, M.; Zhang, H.; Yu, C.; Xu, J.; Wu, H.; Cheang, C.; Jing, Y.; Zhang, W.; Liu, H.; Li, H.; and Kong, T. 2024c. Vision-Language Foundation Models as Effective Robot Imitators. In *International Conference on Learning Representations*, volume 2024, 26703–26721.

Liu, B.; Zhu, Y.; Gao, C.; Feng, Y.; Liu, Q.; Zhu, Y.; and Stone, P. 2023. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In *Advances in Neural Information Processing Systems*, volume 36, 44776–44791.

Ma, N.; Goldstein, M.; Albergo, M. S.; Boffi, N. M.; Vanden-Eijnden, E.; and Xie, S. 2024. Sit: Exploring flow and

diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, 23–40.

Miech, A.; Zhukov, D.; Alayrac, J.-B.; Tapaswi, M.; Laptev, I.; and Sivic, J. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2630–2640.

Octo Model Team; Ghosh, D.; Walke, H.; Pertsch, K.; Black, K.; Mees, O.; Dasari, S.; Hejna, J.; Xu, C.; Luo, J.; Kreiman, T.; Tan, Y.; Sanketi, P.; Vuong, Q.; Xiao, T.; Sadigh, D.; Finn, C.; and Levine, S. 2024. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of Robotics: Science and Systems*.

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.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4195–4205.

Pertsch, K.; Stachowicz, K.; Ichter, B.; Driess, D.; Nair, S.; Vuong, Q.; Mees, O.; Finn, C.; and Levine, S. 2025. Fast: Efficient action tokenization for vision-language-action models. In *Proceedings of Robotics: Science and Systems*.

Qu, D.; Song, H.; Chen, Q.; Yao, Y.; Ye, X.; Ding, Y.; Wang, Z.; Gu, J.; Zhao, B.; Wang, D.; et al. 2025. SpatialVLA: Exploring Spatial Representations for Visual-Language-Action Model. In *Proceedings of Robotics: Science and Systems*.

Sak, H.; Senior, A. W.; Beaufays, F.; et al. 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Interspeech*, volume 2014, 338–342.

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. In *International Conference on Learning Representations*, volume 2025, 92033–92052.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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.

Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Wen, J.; Zhu, Y.; Li, J.; Zhu, M.; Wu, K.; Xu, Z.; Liu, N.; Cheng, R.; Shen, C.; Peng, Y.; et al. 2025. TinyVLA: Toward Fast, Data-Efficient Vision-Language-Action Models for Robotic Manipulation. *IEEE Robotics and Automation Letters*, 10: 3988–3995.

Wu, H.; Jing, Y.; Cheang, C.; Chen, G.; Xu, J.; Li, X.; Liu, M.; Li, H.; and Kong, T. 2024. Unleashing Large-Scale Video Generative Pre-training for Visual Robot Manipulation. In *International Conference on Learning Representations*, volume 2024, 10641–10662.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yang, J.; Tan, R.; Wu, Q.; Zheng, R.; Peng, B.; Liang, Y.; Gu, Y.; Cai, M.; Ye, S.; Jang, J.; et al. 2025a. Magma: A Foundation Model for Multimodal AI Agents. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 14203–14214.

Yang, S.; Li, H.; Chen, Y.; Wang, B.; Tian, Y.; Wang, T.; Wang, H.; Zhao, F.; Liao, Y.; and Pang, J. 2025b. Instructvla: Vision-language-action instruction tuning from understanding to manipulation. *arXiv preprint arXiv:2507.17520*.

Ze, Y.; Zhang, G.; Zhang, K.; Hu, C.; Wang, M.; and Xu, H. 2024. 3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations. In *Proceedings of Robotics: Science and Systems*.

Zhai, X.; Mustafa, B.; Kolesnikov, A.; and Beyer, L. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, 11975–11986.

Zhao, T. Z.; Kumar, V.; Levine, S.; and Finn, C. 2023. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*.

Zhen, H.; Qiu, X.; Chen, P.; Yang, J.; Yan, X.; Du, Y.; Hong, Y.; and Gan, C. 2024. 3D-VLA: A 3D Vision-Language-Action Generative World Model. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, 61229–61245.

Zheng, R.; Liang, Y.; Huang, S.; Gao, J.; Daumé III, H.; Kolobov, A.; Huang, F.; and Yang, J. 2025. TraceVLA: Visual Trace Prompting Enhances Spatial-Temporal Awareness for Generalist Robotic Policies. In *International Conference on Learning Representations*, volume 2025, 54277–54296.