Towards Generalizable Robotic Manipulation in Dynamic Environments

Heng Fang[1], Shangru Li[1], Shuhan Wang[1], Xuanyang Xi[2], Dingkang Liang[1]✉, and Xiang Bai[1]

[1] Huazhong University of Science and Technology, China
[2] Huawei Technologies Co. Ltd, China
✉ Corresponding Author.
{hengfang,dkliang}@hust.edu.cn

Project Page: https://H-EmbodVis.github.io/DOMINO/
Code: https://github.com/H-EmbodVis/DOMINO

Abstract. Vision-Language-Action (VLA) models excel in static manipulation but struggle in dynamic environments with moving targets. This performance gap primarily stems from a scarcity of dynamic manipulation datasets and the reliance of mainstream VLAs on single-frame observations, restricting their spatiotemporal reasoning capabilities. To address this, we introduce **DOMINO**, a large-scale dataset and benchmark for generalizable dynamic manipulation, featuring 35 tasks with hierarchical complexities, over 110K expert trajectories, and a multi-dimensional evaluation suite. Through comprehensive experiments, we systematically evaluate existing VLAs on dynamic tasks, explore effective training strategies for dynamic awareness, and validate the generalizability of dynamic data. Furthermore, we propose **PUMA**, a dynamics-aware VLA architecture. By integrating scene-centric historical optical flow and specialized world queries to implicitly forecast object-centric future states, PUMA couples history-aware perception with short-horizon prediction. Results demonstrate that PUMA achieves state-of-the-art performance, yielding a 6.3% absolute improvement in success rate over baselines. Moreover, we show that training on dynamic data fosters robust spatiotemporal representations that transfer to static tasks.

Keywords: Dynamic Manipulation · Vision-Language-Action Model · Embodied AI

1 Introduction

Recent advancements in Vision-Language-Action (VLA) models have demonstrated significant success in handling manipulation tasks, establishing a foundation for generalizable embodied intelligence [3,13,52,57]. However, most existing VLA models [4,5,26,27] focus on static manipulation, which involves interacting with stationary objects in stable environments. In contrast, dynamic manipulation requires robots to adapt to moving objects and continuous environmental changes. Mastering this capability is essential for deploying robots in complex real-world scenarios, such as operating on active assembly lines or working alongside humans. Despite its importance, dynamic manipulation remains a critical

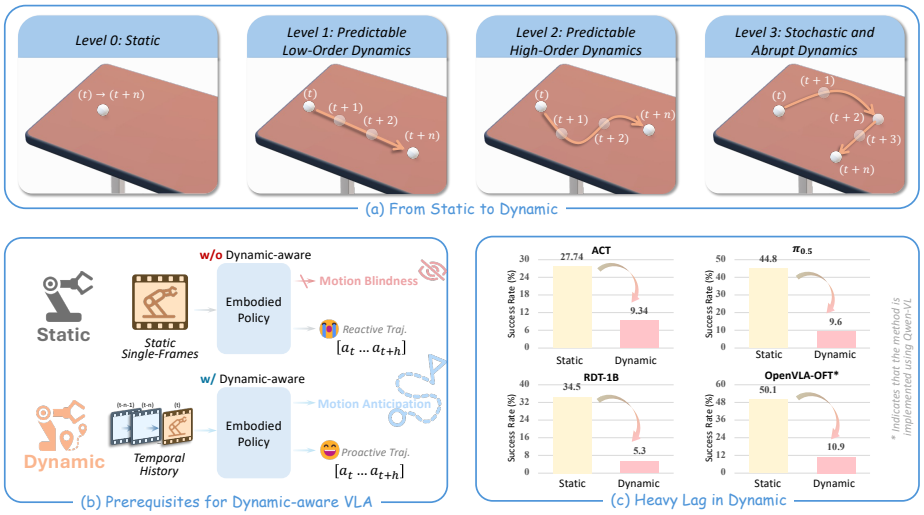


Fig. 1: (a) Illustration of the defined dynamic difficulty levels, progressing from static (Level 0) to stochastic and abrupt dynamics (Level 3). **(b)** Dynamic awareness requires capturing historical context and anticipating future motion. **(c)** Performance of SOTA models degrades when shifting from static to dynamic environments.

yet underexplored frontier. Dynamic manipulation is inherently difficult as it imposes strict requirements on spatiotemporal precision and necessitates the continuous integration of real-time perception with motion prediction. Progress in this domain is hindered by two primary factors: **1)** the scarcity of large-scale dynamic manipulation datasets, and **2)** the limited dynamic perception and motion prediction capabilities of existing mainstream VLA architectures.

High-quality dynamic data is indispensable for achieving generalizable dynamic manipulation. Addressing the current data scarcity is non-trivial due to the inherent complexity of dynamic tasks. Constructing dynamic environments for data collection presents significant challenges, primarily due to the strict spatiotemporal synchronization required between moving targets and robotic actions. Furthermore, acquiring expert demonstrations in such unpredictable settings is difficult, as human teleoperators or scripted policies often struggle to react precisely to continuous environmental changes. Consequently, most existing embodied datasets are confined to stationary tasks [6, 43, 56]. Therefore, developing a scalable, low-cost pipeline to acquire diverse manipulation data in dynamic scenarios is a critical imperative.

To bridge this gap, we introduce a comprehensive benchmark tailored for generalizable dynamic manipulation, termed the **D**ynamic **O**bject **M**an**I**pulatio**N** **O**perations Benchmark (**DOMINO**). Building upon RoboTwin 2.0 [10] and the SAPIEN [61] simulation engine, we develop scalable pipelines for dynamic manipulation dataset generation and closed-loop evaluation. The benchmark comprises 35 diverse dynamic tasks across 5 distinct robot embodiments, spanning from single-arm operations to complex dual-arm collaborations. As shown in Fig. 1(a), these tasks are organized into a three-tiered difficulty hierarchy, progress-

ing from predictable low-order dynamics to high-order nonlinear trajectories and stochastic scenarios with abrupt disturbances. To precisely control task complexity, we parameterize the overall motion speed using a scalar dynamics coefficient α, denoting each setting as *DOMINO@*α. Crucially, these tasks present significant challenges for current VLA models. While existing VLAs excel at static manipulation, they struggle in dynamic environments due to an inherent lack of continuous spatiotemporal reasoning. The precise timing required to interact with moving targets and handle unexpected disturbances often exceeds the capabilities of current architectures, which are typically constrained by static spatial biases. To rigorously assess generalization, our dataset provides over 110K expert trajectories collected under both canonical and domain-randomized settings. Finally, our closed-loop pipeline establishes a multi-dimensional evaluation protocol that extends beyond standard binary success rates.

Evaluations of state-of-the-art models [5, 27, 38, 69] on this benchmark reveal substantial performance degradation in dynamic settings, as shown in Fig. 1(c). We attribute this decline to the reliance of standard VLA architectures on single-frame observations, which limits their dynamic awareness. We suggest that this awareness requires *capturing historical context and anticipating future motion*. Although recent VLAs incorporate world models [9,63,67], they primarily model global scene transitions or robot kinematics, neglecting the individual object dynamics crucial for spatiotemporal anticipation. To address this, we present PUMA as an exploratory attempt to bridge this gap. By integrating historical frames and motion cues to model scene-centric historical dynamics, our method introduces specialized predictive queries to implicitly infer object-centric dynamics for moving targets. This design endows the model with a dynamic understanding of the physical world, enabling anticipatory interactions with moving objects and yielding a 6.3% SR improvement.

In summary, the main contributions are as follows: **1)** We systematically analyze dynamic manipulation, distinguishing its unique spatiotemporal challenges from static paradigms to underscore the need for advancing dynamic embodied intelligence. **2)** We introduce DOMINO, a scalable pipeline for dynamic manipulation dataset generation and closed-loop evaluation. It features diverse robot embodiments, multi-tiered difficulty scaling via a dynamics coefficient, and a comprehensive multi-dimensional metric suite. **3)** We propose PUMA, which integrates historical motion cues to enhance motion anticipation. By employing specialized predictive queries to implicitly infer the future states of moving objects, our approach yields a 6.3% performance improvement over baselines.

2 DOMINO Dataset

2.1 Task Definition

We formulate generalizable dynamic manipulation as a Partially Observable Markov Decision Process (POMDP) [24]. At time step t, the full state $s_t = \{s_t^r, s_t^o\}$ comprises the robot proprioception s_t^r and the physical object state s_t^o. Due to partial observability, the policy relies on an observation history $o_{t-h:t}$, where each observation $o_t = \{I_t, s_t^r\}$ includes high-dimensional visual inputs I_t

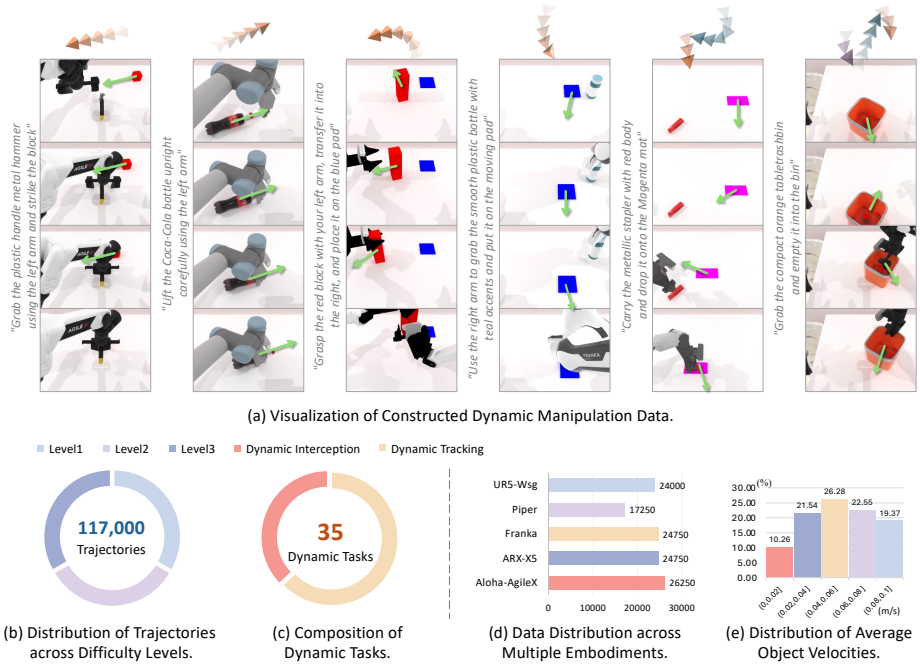


Fig. 2: Dataset Visualization. We present DOMINO dataset of 117,000 dynamic manipulation trajectories, covering 35 distinct tasks across five robot embodiments.

(e.g., RGB-D) and proprioception. The continuous action $\mathbf{a}_t \in \mathcal{A}$ specifies the dual-arm control commands. In dynamic environments, the transition dynamics $\mathcal{T}(s_{t+1}|s_t, \mathbf{a}_t)$ are inherently time-varying, governed by both the independent motion of the object and the interaction dynamics induced by robot contact.

Our objective is to learn a policy $\pi_\phi(\mathbf{a}_t|o_{t-h:t})$ that minimizes the expected finite-horizon cost:

$$J(\phi) = \mathbb{E}\left[\sum_{k=0}^{H-1} \gamma^k \ell(s_{t+k}, \mathbf{a}_{t+k})\right], \tag{1}$$

where $\gamma \in [0, 1)$ is the discount factor, subject to safety constraints. The cost $\ell(\cdot)$ penalizes the spatial discrepancy between the end-effectors and the object, alongside the control effort. To achieve precise spatiotemporal interception, π_ϕ must implicitly anticipate future object states from historical observations.

2.2 Data Construction

To systematically evaluate generalizable dynamic manipulation, we introduce DOMINO. Unlike existing datasets focused on stationary tasks, DOMINO provides a scalable, low-cost pipeline for generating diverse dynamic data. As shown in Fig. 2, it comprises 35 dynamic tasks across five robot embodiments. To facilitate rigorous generalization assessments, we provide over 110K expert trajectories collected under both canonical and domain-randomized settings.

Constructing dynamic environments poses a significant challenge, primarily due to the strict spatiotemporal synchronization required between moving targets and robotic actions. Building upon the SAPIEN [61] physics engine and the RoboTwin 2.0 [10] framework, our data generation pipeline introduces a three-stage spatiotemporal synchronization method to reliably acquire expert demonstrations. First, in the temporal dry-run phase, we randomly sample the manipulation pose of the target object and execute the task in a static environment to record the exact execution time required by the robot arms. Second, in the kinematic back-calculation phase, we reverse-engineer the object's initial spatial position based on the recorded execution time and specified motion trajectory. Third, in the synchronized dynamic execution phase, target objects are instantiated as kinematic bodies within SAPIEN and the robot replays its motion plan alongside the moving target. This ensures stable and predictable motion execution, immune to unintended physical disturbances. To guarantee expert data quality, we implement specialized adaptations for complex objects and enforce strict dynamic task-success criteria during automated generation.

2.3 Data Characteristics

DOMINO categorizes dynamic manipulation through a spatiotemporal task taxonomy, hierarchical motion complexities, and comprehensive evaluation metrics.

Spatiotemporal Task Taxonomy. To decouple the core challenges of dynamic manipulation from task-specific details, we divide the 35 benchmark tasks into two functional categories based on interaction requirements: dynamic interception and dynamic tracking. Dynamic interception focuses on instantaneous target acquisition, where the agent transitions from free space to establish contact with a moving target. This evaluates predictive kinematics and latency compensation. For example, catching a thrown object requires precise trajectory planning to reach an optimal interception point. Isolating this discrete action assesses the agent's spatial precision under strict temporal constraints. In contrast, dynamic tracking involves continuous synchronization, requiring the agent to maintain a consistent spatial relationship with a moving target over a time window $\Delta\tau$. This evaluates real-time closed-loop control and velocity-matching capabilities, such as placing an object into a box on a conveyor belt. Consequently, this formulation assesses agent's capacity for sustained error correction and trajectory adjustment during continuous interactions.

Hierarchical Dynamic Complexity. To evaluate agents across diverse dynamic complexities, ranging from predictable tracking to reactive adaptation, we categorize the motion dynamics $\mathcal{F}$ into three levels, as illustrated in Fig. 1(a):

- Level 1 (Predictable Low-Order Dynamics): Objects move with a constant velocity $\mathbf{v} \sim \mathcal{U}(v_{\min}, v_{\max})$. This zero-curvature trajectory serves as a foundational test for instantaneous state estimation and linear extrapolation.
- Level 2 (Predictable High-Order Dynamics): Trajectories follow a polynomial curve $\mathbf{x}(t) = \sum_{k=0}^{n} \mathbf{b}_k (t/T_{traj})^k$ of degree $n \in [2, 5]$, fit to randomly sampled spatial control points. This introduces variable curvature and acceleration,

requiring the agent to aggregate historical observations to implicitly model the physical dynamics.
- Level 3 (Stochastic and Abrupt Dynamics): Trajectories comprise $s \in [2, 3]$ independent segments of Level 1/2 dynamics, with segment durations drawn from a Dirichlet distribution. Velocity and acceleration are typically discontinuous at segment transitions. This unpredictability evaluates reactive robustness, compelling agent to rely on high-frequency closed-loop feedback.

Comprehensive Evaluation Metrics. To assess robustness across dynamic difficulties, we introduce the benchmark variant DOMINO@α. This variant parameterizes the maximum target speed in meters per second using a scalar coefficient $\alpha \geq 0$. For instance, $\alpha = 0.1$ denotes a maximum speed of 0.1 m/s, and $\alpha = 0$ represents a static setting. Under this setting, we measure the Success Rate (SR), defined as the percentage of episodes that satisfy all task conditions within a time budget $T_{\max}$. As SR alone is insufficient for stochastic environments, we introduce the Manipulation Score (MS), a continuous metric designed to capture execution quality. The MS consists of a base Route Completion (RC) score adjusted by penalty factors. RC quantifies spatial convergence via a progress ratio $\rho = 1 - \|\mathbf{p}_{\text{ee}}^{(T_{end})} - \mathbf{p}_{\text{obj}}^{(T_{end})}\|_2 / \|\mathbf{p}_{\text{ee}}^{(0)} - \mathbf{p}_{\text{obj}}^{(T_{end})}\|_2$, where $\mathbf{p}_{\text{ee}}$ and $\mathbf{p}_{\text{obj}}$ denote positions of the end-effector and target object at initial (0) and final (T_{end}) timesteps. For dual-arm setups, $RC \in [0, 100]$ reflects the maximum progress of either arm, computed as $100 \times \max(\rho_{\text{left}}, \rho_{\text{right}})$, with successful episodes assigned 100. Finally, to penalize unsafe behaviors, the overall MS is calculated by multiplying RC by 0.5 if the target exits the safe workspace or field of view, and by 0.8 upon collision with environmental clutter.

3 Dynamic-Aware VLA

To achieve generalizable dual-arm manipulation in dynamic environments, we propose the **P**redictive **U**nified **M**anipulation **A**rchitecture (**PUMA**) as shown in Fig. 3. Dynamic awareness requires capturing the historical context of objects and anticipating their future motion. Therefore, PUMA couples scene-centric history-aware perception with short-horizon object-centric prediction to satisfy spatiotemporal constraints induced by object dynamics $\mathcal{F}$ within Qwen3-VL [1]. Given an observation history $o_{t-h:t}$ of length $h+1$ and a language instruction l, the model M_θ uses a shared backbone to jointly optimize action policy π_ϕ and an auxiliary future feature predictor ψ_ω. Specifically, it outputs an action chunk $\hat{\mathbf{a}}_{t:t+K-1}$ of length K in a single forward pass [69] alongside auxiliary features $\mathbf{z}_{t+1:t+N}$ of horizon N encoding future object motion. Crucially, the auxiliary future predictor is supervised only during training. This encourages shared representation to anticipate the dynamics $\mathcal{F}$ and effectively regularizes policy without adding computational overhead during inference.

3.1 Scene-Centric Spatiotemporal Dynamics Encoding

To effectively operate in dynamic environments, our model requires a comprehensive understanding of both spatial context and temporal evolution. Therefore, the input space comprises three key components: a long-term language

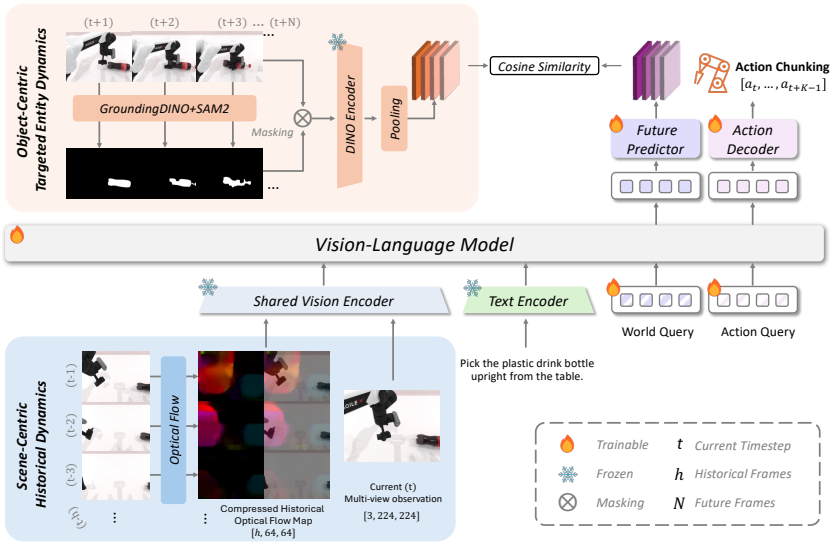


Fig. 3: PUMA processes historical motion flows, current observations, and instructions to encode scene-centric historical dynamics. It employs a dual-query mechanism where Action Queries decode continuous action chunks and World Queries aggregate dynamic representations. During training, world queries are supervised via a similarity loss against future features extracted by DINO to predict object-centric dynamics.

instruction, current multi-view observations, and a historical dynamic context. The language instruction l acts as the overarching task specification, guiding the manipulation policy. Concurrently, the current multi-view images serve as spatial visual prompts, providing dense observations of the immediate workspace. Capturing historical scene dynamics is crucial for reacting to moving objects.

To efficiently encode this dynamics, we introduce a compact dynamic context representation. We sample h historical third-person frames at a fixed stride and apply a spatial compression operator. Instead of directly stacking raw historical frames, which forces the network to implicitly deduce temporal changes, we compute optical flow maps across these compressed frames. Optical flow provides intuitive and explicit motion states, making it significantly easier for the policy to learn dynamic patterns. These flow maps are processed alongside the current multi-view visual input through the Qwen3-VL visual encoder, forming the observation history $o_{t-h:t}$ that supplies the model with explicit dense motion cues, enabling the policy to accurately estimate object motion trends.

3.2 Object-Centric Dynamic Representation

To interact in dynamic environments, the policy must anticipate the future motion of target objects. While existing frameworks [67] forecast scene-level dynamic regions, we posit that generalizable dynamic manipulation requires an object-centric focus, isolating the target's trajectory from irrelevant scene dynamics. To construct supervision for this capability, we sample N future frames $I_{t+1:t+N}$ at a fixed interval during training and isolate the target object using a

frozen grounding module [51]. Specifically, the manipulated object parsed from the language instruction serves as a text prompt p for GroundingDINO [37], which generates a bounding box, then processed by SAM2 [49] to yield a segmentation mask. Let $\mathcal{B}(I, p)$ denote the binary mask, $\mathcal{E}(I)$ the frozen DINO patch-token encoder, and $\mathcal{P}(\cdot, \cdot)$ masked average pooling. The object-centric future feature $\mathbf{f}_{t+i}$ is computed as:

$$\mathbf{f}_{t+i} = \mathcal{P}(\mathcal{E}(I_{t+i}), \mathcal{B}(I_{t+i}, p)), \quad i = 1, \ldots, N. \tag{2}$$

To model these future states, we introduce N learnable world queries that aggregate the spatiotemporal context to predict the target's future representations within the latent space. We then optimize these predictions by enforcing a similarity loss against the extracted ground-truth DINO features. This explicit supervision forces the latent world representation to capture and anticipate the underlying object dynamics. This supervision is applied only during training, requiring no future frames at inference.

3.3 Training Strategy

We train the unified model M_θ end-to-end using supervised behavioral cloning combined with an auxiliary future-feature prediction objective. Each training tuple comprises $\{o_{t-h:t}, l, \mathbf{f}_{t+1:t+N}, \mathbf{a}^*_{t:t+K-1}\}$. The action policy is supervised via an ℓ_1 regression loss over the predicted action chunk:

$$\mathcal{L}_{action} = \frac{1}{K} \sum_{i=0}^{K-1} \|\hat{\mathbf{a}}_{t+i} - \mathbf{a}^*_{t+i}\|_1. \tag{3}$$

Simultaneously, the auxiliary future predictor is optimized by minimizing the cosine distance between the predicted representations $\mathbf{z}_{t+i}$ and the ground-truth object-centric features $\mathbf{f}_{t+i}$:

$$\mathcal{L}_{world} = \frac{1}{N} \sum_{i=1}^{N} \left(1 - \frac{\mathbf{z}_{t+i}^\top \mathbf{f}_{t+i}}{\|\mathbf{z}_{t+i}\|_2 \, \|\mathbf{f}_{t+i}\|_2} \right). \tag{4}$$

Overall training objective is formulated as a weighted sum of two components:

$$\mathcal{L}_{total} = \mathcal{L}_{action} + \lambda \, \mathcal{L}_{world}, \tag{5}$$

where λ is a balancing hyperparameter controlling influence of dynamics task.

4 Experiment

All VLA models are trained on NVIDIA A100 GPUs, while data generation and evaluation are performed on NVIDIA RTX GPUs. Appendix provides additional implementation details and hyperparameters. To balance broad experimental coverage with limited compute, unless otherwise stated, all experiments are conducted in clean setting (plain backgrounds, no domain randomization) with the Aloha-AgileX robot, using a training mixture of 35 dynamic tasks under Level 1 dynamics with dynamic coefficient $\alpha = 0.1$.

4.1 Experimental Setup

Benchmarks. We primarily evaluate on our proposed DOMINO@0.1 benchmark, reporting the Success Rate (SR) and Manipulation Score (MS). Additionally, we provide selected results in static environments using RoboTwin 2.0 [10].

Baselines. We evaluate PUMA against the standard policy learning framework ACT [69] and several state-of-the-art VLA models, including OpenVLA [27], OpenVLA-OFT [26], RDT [38], π_0 [5], $\pi_{0.5}$ [4], π_0-FAST [45], VLA-Adapter [57], InternVLA-M1 [12], and Qwen3-VL-based VLAs [11]. For a fair comparison, all baselines are fine-tuned on proposed DOMINO dataset. Note that while the VLA models are fine-tuned across all tasks, ACT is fine-tuned on a per-task basis.

4.2 Challenges of the Proposed DOMINO Dataset

Table 1: Quantitative evaluation of VLA models in static and our proposed DOMINO@0.1 under Level 1. $X \rightarrow Y$ denotes training in environment X and testing in Y (S: static, D: dynamic), with ZS representing zero-shot evaluation and FT indicating fine-tuning. Subscripts denote performance Δ (ZS vs. Static; FT vs. ZS).

Simulation Task	**ACT [69]**			**OpenVLA-OFT [26]**			**$\pi_{0.5}$ [4]**		
	Static S→S	ZS S→D	FT D→D	Static S→S	ZS S→D	FT D→D	Static S→S	ZS S→D	FT D→D
Adjust Bottle	97%	37%	65%	47%	16%	58%	92%	12%	52%
Click Alarmclock	29%	3%	8%	57%	9%	6%	48%	8%	14%
Grab Roller	95%	24%	23%	47%	24%	28%	98%	6%	10%
Handover Block	43%	0%	11%	3%	3%	9%	2%	0%	1%
Handover Mic	82%	13%	23%	93%	8%	25%	63%	7%	5%
Move Can Pot	23%	0%	26%	34%	29%	29%	14%	5%	7%
Move Pillbottle Pad	1%	0%	1%	1%	0%	1%	20%	5%	6%
Place Bread Skillet	9%	5%	12%	4%	1%	1%	39%	5%	9%
Place Fan	0%	0%	1%	3%	0%	1%	30%	1%	2%
Place Object Basket	18%	19%	21%	19%	16%	18%	57%	6%	11%
.....(35 tasks)									
Press Stapler	31%	2%	4%	35%	2%	5%	35%	1%	6%
Rotate QRcode	3%	0%	0%	10%	8%	11%	54%	2%	4%
Scan Object	1%	2%	2%	0%	0%	4%	7%	0%	1%
Shake Bottle	74%	18%	14%	63%	12%	17%	95%	19%	33%
Shake Bottle Horiz.	63%	23%	18%	57%	19%	31%	97%	24%	42%
Average (%)	27.7	$6.5_{-21.2}$	$9.4_{+2.9}$	17.5	$6.7_{-10.8}$	$9.1_{+2.4}$	44.8	$7.5_{-37.3}$	$9.6_{+2.1}$

To investigate the challenges of dynamic environments, we evaluate representative VLA architectures in both static and dynamic settings. As shown in Tab. 1, these models exhibit satisfactory performance in static scenarios (S→S), but their performance degrades significantly in dynamic environments (S→D). Under identical task settings with moving targets, the average success rate drops drastically. For instance, the performance of $\pi_{0.5}$ [4] falls from 44.8% to 7.5%.

Furthermore, we explore whether fine-tuning these baselines with dynamic data

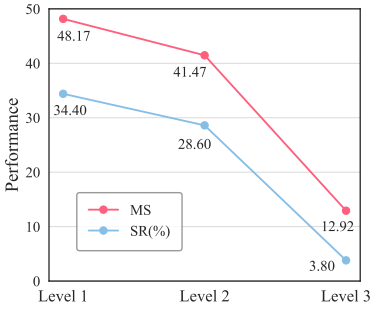


Fig. 4: Performance degradation of the ACT model across three dynamic complexity (Level 1-3).

Table 2: Results of explicitly modeling future trajectories on Level 1 DOMINO@0.1.

Method	Settings	w/ GT Traj.	SR (%)↑	MS↑
OpenVLA-OFT [26]	Static	×	17.51	17.51
	Dynamic	×	9.06	24.06
	Dynamic	✓	10.33	32.00

(D→D) can bridge this gap. We observe only marginal improvements, with average success rates increasing by less than 3%. We attribute this bottleneck to fundamental architectural limitations rather than mere data distribution shifts. Standard VLA architectures rely on single-frame observations, inherently limiting their dynamic awareness. We argue that effective dynamic manipulation requires *capturing historical object context and anticipating future motion*. The current VLA paradigm lacks these essential dynamic modeling capabilities, resulting in sub-optimal dynamic manipulation performance.

While our primary experiments focus on Level 1 dynamics, we also investigate the impact of higher dynamic complexities. We train ACT models per-task on five tasks and evaluate them across all three dynamic levels. As illustrated in Fig. 4, performance deteriorates rapidly as complexity increases. Level 2 and Level 3 environments are significantly more challenging than Level 1, highlighting the need for stronger dynamic modeling architectures in future research.

***Finding 1**: Dynamic manipulation presents a challenging new frontier.* The transition from static to dynamic environments introduces complex spatiotemporal challenges that fundamentally degrade the reliability of existing paradigms. It represents a distinct and demanding domain that cannot be solved by merely scaling static manipulation approaches.

To validate this hypothesis and isolate the performance bottleneck of current VLAs, we conduct an oracle experiment, as detailed in Tab. 2. We introduce an auxiliary dynamic ground-truth encoder, implemented as an MLP. This encoder takes the target object's future pose window combined with a continuous trajectory parameter vector as input to generate a dynamic conditional representation. During the forward pass, this conditional vector is concatenated with the action queries and passed through a subsequent MLP layer. Experimental results reveal a nuanced behavior: while the explicit introduction of future trajectories yields only a marginal improvement in overall SR, it significantly boosts MS. Qualitative analysis of the evaluation rollouts indicates that the policy successfully acquires and tracks the correct trajectory but exhibits control jitter and temporal inconsistency during the actual manipulation phase. We attribute this to the lack of historical frame observations; without historical context, the model fails to comprehend the target's underlying physical dynamics. Instead, the policy overfits to naive trajectory following, which inadvertently interferes with closed-loop manipulation learning. However, in successful episodes, the introduction of ground-truth trajectories yields exceptionally high manipulation quality that approaches expert demonstrations, confirming the potential of future spatial cues if appropriately grounded in physical context.

Table 3: Comparison with SOTA methods on DOMINO@0.1 (Level 1 dynamics, clean setting, Aloha-AgileX robot). All VLA models are fine-tuned on 35 dynamic tasks.

Method	Venue	LLM	SR (%)↑	MS↑
OpenVLA [27]	CoRL 24	Prismatic	1.54	6.10
RDT-1B [38]	ICLR 25	DiT	5.34	17.71
π_0 [5]	RSS 25	PaliGemma	8.17	23.96
$\pi_{0.5}$ [4]	CoRL 25	PaliGemma	9.63	26.17
InternVLA-M1 [12]	arXiv 25	Qwen2.5-VL	5.40	27.57
Isaac-GR00T [3]	arXiv 25	Qwen3-VL*	6.10	28.60
VLA-Adapter [57]	AAAI 26	Qwen2.5-VL	4.40	24.31
π_0-FAST [45]	RSS 25	PaliGemma	3.54	20.87
		Qwen3-VL*	5.74	20.66
OpenVLA-OFT [26]	RSS 25	Prismatic	9.06	24.06
		Qwen3-VL*	10.86	30.49
PUMA (OURS)	-	Qwen3-VL	**17.20**	**34.97**

* Methods with Qwen3-VL backbones are our re-implementations for fair comparison.

***Finding 2**: Naive future trajectory injection is insufficient for dynamic manipulation.* While providing future spatial cues improves motion tracking, the absence of historical observations prevents the model from understanding physical dynamics, leading to control instability during manipulation. Effective dynamic manipulation requires a holistic integration of both historical context and future anticipation.

4.3 Effectiveness of Dynamic-Aware VLA

Building upon the insight that effective dynamic manipulation requires both historical context and future anticipation, we evaluate PUMA as a solution to these spatiotemporal challenges. As shown in Tab. 3, PUMA demonstrates a performance advantage over SOTA VLA models on dynamic manipulation tasks.

Specifically, PUMA achieves the highest average success rate of 17.20%, substantially outperforming recent strong baselines such as OpenVLA-OFT (Qwen3-VL-based) [11] (10.86%) and $\pi_{0.5}$ [4] (9.63%). Furthermore, our method attains a peak Manipulation Score of 34.97, indicating a higher quality of interaction with moving targets. We attribute this performance leap to the explicit integration of historical observation frames and the introduction of the auxiliary future feature predictor ψ_ω. By forcing shared representation to anticipate object dynamics during training, model learns to infer future states of moving objects during inference. This design effectively mitigates the dynamic awareness bottleneck identified in previous section.

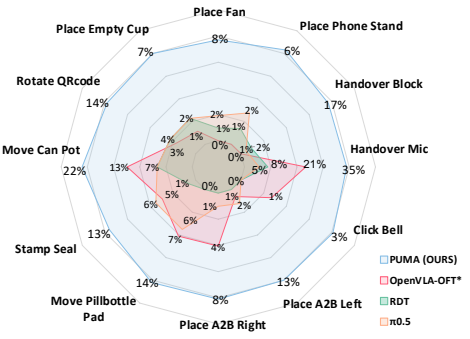


Fig. 5: PUMA performs significantly better than other baselines on difficult tasks.

To provide a detailed task-level comparison, we visualize the performance breakdown across individual manipulation tasks in Fig. 5. Notably, PUMA yields

Table 4: Task-type breakdown on Level 1 DOMINO@0.1. DI: Dynamic Interception, DT: Dynamic Tracking.

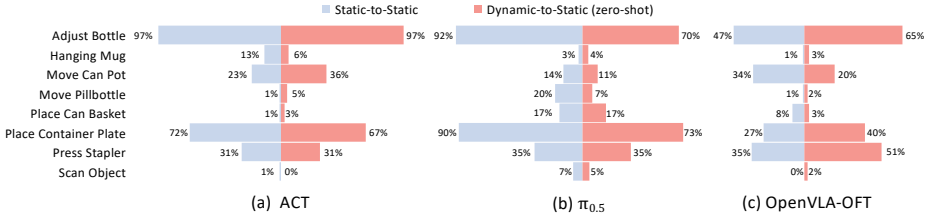
Method	DI		DT	
	SR (%)	MS	SR (%)	MS
Open.-OFT [26]	14.3	34.0	2.9	12.3
$\pi_{0.5}$ [4]	11.8	33.6	7.0	17.3
PUMA	**24.2**	**47.9**	**8.9**	**19.6**

Table 5: Cross-level evaluation on the 10-task subset. PUMA adapts a Level 1 checkpoint with LoRA.

Method	Level 2		Level 3	
	SR (%)	MS	SR (%)	MS
$\pi_{0.5}$ [4]	7.9	20.76	1.6	13.39
PUMA	**10.5**	**30.28**	**4.6**	**20.16**

Table 6: Comparison between co-training and dynamic data training on Level 1 DOMINO@0.1. * indicates our reproduction of OpenVLA-OFT [11] with the same Qwen3-VL backbone for fairness. To isolate gains from data mixing, PUMA is evaluated with prediction horizon $N = 2$.

Method	Static	Dynamic	SR (%)↑	MS↑
OpenVLA-OFT* [11]	×	✓	10.86	30.49
	✓	✓	12.89$_{+2.03}$	30.80$_{+0.31}$
PUMA	×	✓	14.80	32.74
	✓	✓	19.71$_{+4.91}$	37.76$_{+5.02}$



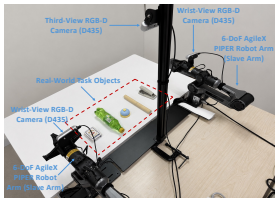
The model trained on DOMINO (coral) demonstrates generalization ability when tested in zero-shot scenarios to static scenes, and achieves comparable results to static data(blue) in some tasks.

substantial improvements on highly challenging dynamic tasks where baseline methods struggle and achieve only marginal success rates. These pronounced performance gains underscore robustness of our approach in handling complex object dynamics. Overall, the quantitative results validate that our unified, dynamics-aware design represents an effective step toward generalizable manipulation in dynamic environments.

Tab. 4 shows that the two baselines have opposing strengths: OpenVLA-OFT favors DI (14.3% vs. 2.9% SR on DT) while $\pi_{0.5}$ is stronger on DT (7.0% vs. 11.8% on DI); PUMA improves both, with DI gains from the auxiliary predictor estimating interception points and DT gains from optical flow enabling closed-loop tracking. This advantage holds under higher complexity (Tab. 5): PUMA outperforms $\pi_{0.5}$ at both Level 2 (10.5% vs. 7.9%) and Level 3 (4.6% vs. 1.6% SR), confirming that temporal reasoning from simpler dynamics transfers to more complex motion patterns. Detailed analysis is provided in Appendix.

4.4 The Role of Dynamic Data in Generalization

To investigate the generalization capability of dynamic data, we evaluate the zero-shot performance of policies trained exclusively on dynamic datasets when deployed in static environments. As shown in the bar charts of Tab. 6, dynamic-



Method	Bell	Roller	Bottle	Card	Stapler	Avg
ACT	5	10	5	5	10	7
RDT-1B	10	5	10	0	15	8
$\pi_{0.5}$	15	20	20	25	40	24
PUMA	**50**	**45**	**35**	**25**	**55**	**42**

Fig. 6: Left: Real-world setup with two 6-DoF AgileX Piper arms. **Right:** Real-world SR (%) on five dynamic tasks. Each entry averages over 20 trials.

Method	Latency (ms)	Freq. (Hz)
OpenVLA	166.7	6.0
Open.-OFT	76.9	13.0
π_0	106.5	9.4
$\pi_{0.5}$	103.0	9.7
PUMA	103.7	9.6

Table 7: Inference latency on a single RTX 4090 GPU.

trained models achieve comparable performance to their static-trained counterparts on certain tasks. For instance, using the OpenVLA-OFT baseline, the dynamic-trained policy outperforms the static-trained policy on Adjust Bottle (65% vs. 47%) and Place Container Plate (40% vs. 27%) tasks, while maintaining parity on Adjust Bottle task (97%) under ACT baseline. These results suggest that dynamic training can supplement static training, yielding transfer benefits on tasks where the required manipulation primitives partially overlap.

Furthermore, we evaluate co-training policies on mixed static and dynamic datasets. As detailed in Tab. 6, this hybrid training strategy yields significant performance improvements in dynamic environments. Specifically, for PUMA, incorporating static data increases the overall SR by 4.91% (from 14.80% to 19.71%) and the MS by 5.02 compared to training on dynamic data alone. We hypothesize that this synergy arises because static data provides stable structural priors for foundational manipulation, while dynamic data introduces the spatiotemporal variations necessary for reactive dexterity.

***Finding 3**: Dynamic data fosters generalizable representations.* Exposure to dynamic interactions mitigates overfitting to static positional biases, encouraging the policy to learn robust spatiotemporal representations. This facilitates effective zero-shot transfer to static environments and, when co-trained with static data, maximizes dynamic manipulation performance by combining stable foundational priors with reactive dexterity.

4.5 Real-World Robot Experiments

We validate sim-to-real transfer on the bimanual AgileX Piper platform illustrated in Fig. 6 (left), whose hardware configuration matches our simulation. We select five dynamic tasks from DOMINO (Click Bell, Grab Roller, Lift Bottle, Move Playingcard Away, Press Stapler) and collect 50 teleoperated demonstrations per task. All methods are trained on this shared dataset; PUMA is LoRA fine-tuned from its simulation checkpoint. As reported in Fig. 6 (right), PUMA achieves 42% average SR, compared with 24% for $\pi_{0.5}$ and below 10% for ACT and RDT, which lack dynamic awareness. These results confirm that dynamic awareness acquired in simulation transfers to real hardware via lightweight LoRA adaptation. Real-robot rollout visualizations are provided in Appendix.

Table 8: Ablation of core components and future prediction steps (N) on Level 1 DOMINO@0.1. We evaluate the impact of historical representations (Hist. Rep.), auxiliary future prediction (Aux. Pred.), and the prediction horizon.

Method	Components			SR (%)↑	MS↑
	Hist. Rep.	Aux. Pred.	Steps (N)		
Baseline	×	×	-	10.86	30.49
+ Hist. Flow	✓	×	-	11.71	31.02
+ Hist. Flow	✓	✓	2	14.80	32.74
+ Hist. Frames	✓	✓	2	8.15	28.62
+ Hist. Flow	✓	✓	4	**17.20**	**34.97**

4.6 Inference Latency and Control Frequency

We benchmark end-to-end inference latency on a single RTX 4090 (batch size 1, bf16, averaged over 500 queries after 50 warm-ups). As shown in Tab. 7, PUMA runs at 103.7 ms per step (9.6 Hz), matching $\pi_{0.5}$. The Qwen3-VL backbone accounts for 98.3 ms (94.7%); optical-flow computation adds 3.45 ms (3.3%) and preprocessing 2.0 ms (2.0%). The dynamic-awareness modules add negligible overhead, and action chunking amortizes each query over multiple control steps.

4.7 Ablation Study

We conduct an ablation study to evaluate our proposed modules, as detailed in Tab. 8. First, providing explicit motion cues via optical flow (+ Hist. Flow) improves the baseline SR from 10.86% to 11.71%. Adding the auxiliary future prediction task at $N = 2$ (+ Hist. Flow with Aux. Pred.) further boosts the SR to 14.80%, confirming that anticipating future states effectively regularizes the action policy. Crucially, replacing optical flow with raw historical frames (+ Hist. Frames) degrades the SR to 8.15%, demonstrating that implicitly deducing temporal dynamics from raw frames is suboptimal compared to utilizing explicit flow representations. Finally, extending the prediction horizon to $N = 4$ achieves the best performance (17.20% SR, 34.97 MS), suggesting that a longer anticipation horizon enables a more robust understanding of future trajectories.

5 Related Work

5.1 Vision-Language-Action Model

Vision-Language-Action models [2, 6, 50, 53, 65, 74] have transformed robotic manipulation by directly mapping multimodal inputs and instructions into control commands, and this paradigm has been extended to autonomous driving [17, 18] and multi-task coordination [31]. Recent architectures [3–5, 12, 26, 27, 38] leverage mechanisms like diffusion policies to achieve robust zero-shot adaptability. However, their application in dynamic real-world environments is bottlenecked by inadequate spatiotemporal modeling. Regarding temporal modeling, the high computational cost of processing multi-frame observations forces most VLAs to operate as memoryless single-frame policies. This precludes the extraction of continuous dynamics. While recent efforts incorporate temporal contexts [8, 14, 16, 19, 22, 44, 55] or memory mechanisms for long-horizon tasks [21, 28, 34, 36, 52, 70],

they primarily track task progression. They fail to perform high-frequency motion estimation required for dynamic manipulation. In terms of spatial representation, standard VLAs assume static environments and neglect the continuous motion of manipulated objects [48], despite progress in 3D spatial understanding [30,33,60]. Approaches like ReconVLA [54] allocate visual attention to target objects. DreamVLA [67] integrated dream query to predict global scene transitions, and scene-level world models [32,71,72] forecast future 3D states. However, these methods lack the fine-grained object-centric dynamic modeling essential for reactive planning. Furthermore, existing dynamic manipulation methods [58,68] are typically restricted to simplified motion tasks and fail to capture real-world dynamic complexity. To bridge these gaps, we introduce spatiotemporal designs to effectively tackle dynamic manipulation tasks.

5.2 Datasets and Benchmarks for Robotic Manipulation

Robot learning benchmarks primarily encompass real-world and simulation environments. While real-world platforms [7, 25, 43, 56, 59, 62] enable standardized training and evaluation on physical robots, they are often hindered by limited reproducibility, hardware variations, and safety constraints. Consequently, simulated closed-loop environments remain indispensable for scalable and reliable policy evaluation [15, 23, 39, 41, 46, 64, 73]. In simulation, foundational benchmarks like RLBench [20] and CALVIN [40] established multi-task, long-horizon evaluation. Subsequent platforms address diverse challenges: LIBERO [35] for lifelong knowledge transfer, and RoboCasa [42] for generative task scaling. Furthermore, The Colosseum [47] and SIMPLER [29] evaluate out-of-distribution robustness, while RoboTwin2.0 [10] and VLABench [66] explore bimanual coordination and cognitive reasoning. However, existing simulation benchmarks fundamentally rely on a static world assumption where state transitions are strictly robot-driven, failing to assess the manipulation of independently moving targets. To address this, we introduces a comprehensive benchmark designed to evaluate generalizable dual-arm manipulation capabilities in dynamic environments.

6 Conclusion

Generalizable robotic manipulation in dynamic environments remains a critical yet underexplored challenge in embodied AI. We introduce DOMINO, a scalable benchmark with hierarchical dynamic complexities, and show that existing VLAs suffer severe performance degradation when targets move, exposing a lack of spatiotemporal reasoning in current architectures. To address this, we propose PUMA, which integrates historical optical flow with object-centric future prediction to anticipate target motion and achieve state-of-the-art performance. Our analysis further reveals that dynamic training yields representations that generalize to static tasks, confirming the complementarity of the two domains.

Acknowledgments

This work was supported in part by NSFC (62225603, 623B2038) and in part by Hubei Science and Technology Major Project (2024BAA007).

References

1. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025) 6
2. Belkhale, S., Ding, T., Xiao, T., Sermanet, P., Vuong, Q., Tompson, J., Chebotar, Y., Dwibedi, D., Sadigh, D.: Rt-h: Action hierarchies using language. In: Proceedings of Robotics: Science and Systems (2024) 14
3. Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L., Fang, Y., Fox, D., Hu, F., Huang, S., et al.: Gr00t n1: An open foundation model for generalist humanoid robots. arXiv preprint arXiv:2503.14734 (2025) 1, 11, 14
4. Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M.R., Finn, C., Fusai, N., Galliker, M.Y., et al.: \$\pi_{0.5}$: a vision-language-action model with open-world generalization. In: 9th Annual Conference on Robot Learning (2025) 1, 9, 11, 12, 14
5. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: \π_0: A vision-language-action flow model for general robot control. In: Proceedings of Robotics: Science and Systems (2025) 1, 3, 9, 11, 14
6. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al.: Rt-1: Robotics transformer for real-world control at scale. Robotics: Science and Systems XIX (2023) 2, 14
7. Bu, Q., Cai, J., Chen, L., Cui, X., Ding, Y., Feng, S., Gao, S., He, X., Hu, X., Huang, X., et al.: Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2025) 15
8. Bu, Q., Yang, Y., Cai, J., Gao, S., Ren, G., Yao, M., Luo, P., Li, H.: Univla: Learning to act anywhere with task-centric latent actions. In: Proceedings of Robotics: Science and Systems (2025) 14
9. Cen, J., Yu, C., Yuan, H., Jiang, Y., Huang, S., Guo, J., Li, X., Song, Y., Luo, H., Wang, F., et al.: Worldvla: Towards autoregressive action world model. arXiv preprint arXiv:2506.21539 (2025) 3
10. Chen, T., Chen, Z., Chen, B., Cai, Z., Liu, Y., Li, Z., Liang, Q., Lin, X., Ge, Y., Gu, Z., et al.: Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. arXiv preprint arXiv:2506.18088 (2025) 2, 5, 9, 15
11. Community, S.: Starvla: A lego-like codebase for vision-language-action model developing. arXiv preprint arXiv:2604.05014 (2026) 9, 11, 12
12. Contributors, I.M.: Internvla-m1: A spatially guided vision-language-action framework for generalist robot policy. arXiv preprint arXiv:2510.13778 (2025) 9, 11, 14
13. Cui, C., Ding, P., Song, W., Bai, S., Tong, X., Ge, Z., Suo, R., Zhou, W., Liu, Y., Jia, B., Zhao, H., Huang, S., Wang, D.: Openhelix: A short survey, empirical analysis, and open-source dual-system vla model for robotic manipulation. arXiv preprint arXiv:2505.03912 (2025) 1

14. Driess, D., Xia, F., Sajjadi, M.S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al.: Palm-e: An embodied multimodal language model. In: International Conference on Machine Learning. pp. 8469–8488. PMLR (2023) 14
15. Ehsani, K., Han, W., Herrasti, A., VanderBilt, E., Weihs, L., Kolve, E., Kembhavi, A., Mottaghi, R.: Manipulathor: A framework for visual object manipulation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4497–4506 (2021) 15
16. Fan, C., Jia, X., Sun, Y., Wang, Y., Wei, J., Gong, Z., Zhao, X., Tomizuka, M., Yang, X., Yan, J., et al.: Interleave-vla: Enhancing robot manipulation with interleaved image-text instructions. arXiv preprint arXiv:2505.02152 (2025) 14
17. Fu, H., Zhang, D., Zhao, Z., Cui, J., Liang, D., Zhang, C., Zhang, D., Xie, H., Wang, B., Bai, X.: Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2025) 14
18. Fu, H., Zhang, D., Zhao, Z., Cui, J., Xie, H., Wang, B., Chen, G., Liang, D., Bai, X.: Minddrive: A vision-language-action model for autonomous driving via online reinforcement learning. arXiv preprint arXiv:2512.13636 (2025) 14
19. Guan, Y., Yin, L., Liang, D., Ju, J., Luo, Z., Luan, J., Liu, Y., Bai, X.: Video streaming thinking: Videollms can watch and think simultaneously. arXiv preprint arXiv:2603.12262 (2026) 14
20. James, S., Ma, Z., Rovick Arrojo, D., Davison, A.J.: Rlbench: The robot learning benchmark & learning environment. IEEE Robotics and Automation Letters (2020) 15
21. Jang, H., Yu, S., Kwon, H., Jeon, H., Seo, Y., Shin, J.: Contextvla: Vision-language-action model with amortized multi-frame context. arXiv preprint arXiv:2510.04246 (2025) 14
22. Jiang, Y., Gupta, A., Zhang, Z., Wang, G., Dou, Y., Chen, Y., Fei-Fei, L., Anandkumar, A., Zhu, Y., Fan, L.: Vima: Robot manipulation with multimodal prompts. In: International Conference on Machine Learning. pp. 14975–15022. PMLR (2023) 14
23. Jiang, Z., Xie, Y., Lin, K., Xu, Z., Wan, W., Mandlekar, A., Fan, L.J., Zhu, Y.: Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 16923–16930. IEEE (2025) 15
24. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artificial intelligence **101**(1-2), 99–134 (1998) 3
25. Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., Nasiriany, S., Srirama, M.K., Chen, L.Y., Ellis, K., et al.: Droid: A large-scale in-the-wild robot manipulation dataset. In: Proceedings of Robotics: Science and Systems (2024) 15
26. Kim, M.J., Finn, C., Liang, P.: Fine-tuning vision-language-action models: Optimizing speed and success. In: Proceedings of Robotics: Science and Systems (2025) 1, 9, 10, 11, 12, 14
27. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E.P., Sanketi, P.R., Vuong, Q., et al.: Openvla: An open-source vision-language-action model. In: 8th Annual Conference on Robot Learning 1, 3, 9, 11, 14
28. Koo, M., Choi, D., Kim, T., Lee, K., Kim, C., Seo, Y., Shin, J.: Hamlet: Switch your vision-language-action model into a history-aware policy. In: ICLR (2026) 14

29. Li, X., Hsu, K., Gu, J., Pertsch, K., Mees, O., Walke, H.R., Fu, C., Lunawat, I., Sieh, I., Kirmani, S., Levine, S., Wu, J., Finn, C., Su, H., Vuong, Q., Xiao, T.: Evaluating real-world robot manipulation policies in simulation. In: Conference on Robot Learning (2024) 15
30. Liang, D., Feng, T., Zhou, X., Zhang, Y., Zou, Z., Bai, X.: Parameter-efficient fine-tuning in spectral domain for point cloud learning. IEEE transactions on pattern analysis and machine intelligence (2025) 15
31. Liang, D., Zhang, C., Xu, X., Ju, J., Luo, Z., Bai, X.: Cook and clean together: Teaching embodied agents for parallel task execution. In: Proceedings of the AAAI Conference on Artificial Intelligence (2026) 14
32. Liang, D., Zhang, D., Zhou, X., Tu, S., Feng, T., Li, X., Zhang, Y., Du, M., Tan, X., Bai, X.: Unifuture: A 4d driving world model for future generation and perception. In: IEEE International Conference on Robotics and Automation (2026) 15
33. Liang, D., Zhou, X., Xu, W., Zhu, X., Zou, Z., Ye, X., Tan, X., Bai, X.: Pointmamba: A simple state space model for point cloud analysis. In: Advances in Neural Information Processing Systems (2024) 15
34. Lin, M., Ding, P., Wang, S., Zhuang, Z., Liu, Y., Tong, X., Song, W., Lyu, S., Huang, S., Wang, D.: Hif-vla: Hindsight, insight and foresight through motion representation for vision-language-action models. In: CVPR (2026) 14
35. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. Advances in Neural Information Processing Systems **36**, 44776–44791 (2023) 15
36. Liu, C., Zhang, J., Li, C., Zhou, Z., Wu, S., Huang, S., Duan, H.: Ttf-vla: Temporal token fusion via pixel-attention integration for vision-language-action models. In: AAAI (2026) 14
37. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In: ECCV (2024) 8
38. Liu, S., Wu, L., Li, B., Tan, H., Chen, H., Wang, Z., Xu, K., Su, H., Zhu, J.: Rdt-1b: A diffusion foundation model for bimanual manipulation. In: ICLR (2025) 3, 9, 11, 14
39. Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., et al.: Isaac gym: High performance gpu based physics simulation for robot learning. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2) 15
40. Mees, O., Hermann, L., Rosete-Beas, E., Burgard, W.: Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. IEEE Robotics and Automation Letters **7**(3), 7327–7334 (2022) 15
41. Mu, T., Ling, Z., Xiang, F., Yang, D.C., Li, X., Tao, S., Huang, Z., Jia, Z., Su, H.: Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2) 15
42. Nasiriany, S., Maddukuri, A., Zhang, L., Parikh, A., Lo, A., Joshi, A., Mandlekar, A., Zhu, Y.: Robocasa: Large-scale simulation of everyday tasks for generalist robots. In: Proceedings of Robotics: Science and Systems (2024) 15
43. O'Neill, A., Rehman, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., Mandlekar, A., Jain, A., et al.: Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. pp. 6892–6903. IEEE (2024) 2, 15

44. Patratskiy, M.A., Kovalev, A.K., Panov, A.I.: Spatial traces: Enhancing vla models with spatial-temporal understanding. Optical Memory and Neural Networks **34**(Suppl 1), S72–S82 (2025) 14
45. Pertsch, K., Stachowicz, K., Ichter, B., Driess, D., Nair, S., Vuong, Q., Mees, O., Finn, C., Levine, S.: Fast: Efficient action tokenization for vision-language-action models. In: Proceedings of Robotics: Science and Systems (2025) 9, 11
46. Puig, X., Undersander, E., Szot, A., Cote, M.D., Yang, T.Y., Partsey, R., Desai, R., Clegg, A.W., Hlavac, M., Min, S.Y., et al.: Habitat 3.0: A co-habitat for humans, avatars and robots. In: ICLR (2024) 15
47. Pumacay, W., Singh, I., Duan, J., Krishna, R., Thomason, J., Fox, D.: The colosseum: A benchmark for evaluating generalization for robotic manipulation. In: Proceedings of Robotics: Science and Systems (2024) 15
48. Qiu, W., Huang, T., Feng, A., Ying, R.: Efficient long-horizon vision-language-action models via static-dynamic disentanglement. arXiv preprint arXiv:2602.03983 (2026) 15
49. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollár, P., Feichtenhofer, C.: Sam 2: Segment anything in images and videos. In: ICLR (2025) 8
50. Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S.G., Novikov, A., Barth-maron, G., Giménez, M., Sulsky, Y., Kay, J., Springenberg, J.T., et al.: A generalist agent. Transactions on Machine Learning Research 14
51. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., Zeng, Z., Zhang, H., Li, F., Yang, J., Li, H., Jiang, Q., Zhang, L.: Grounded sam: Assembling open-world models for diverse visual tasks (2024) 8
52. Shi, H., Xie, B., Liu, Y., Sun, L., Liu, F., Wang, T., Zhou, E., Fan, H., Zhang, X., Huang, G.: Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In: ICLR (2026) 1, 14
53. Shridhar, M., Manuelli, L., Fox, D.: Cliport: What and where pathways for robotic manipulation. In: Conference on robot learning. pp. 894–906. PMLR (2022) 14
54. Song, W., Zhou, Z., Zhao, H., Chen, J., Ding, P., Yan, H., Huang, Y., Tang, F., Wang, D., Li, H.: Reconvla: Reconstructive vision-language-action model as effective robot perceiver. In: AAAI (2026) 15
55. Team, O.M., Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Kreiman, T., Xu, C., et al.: Octo: An open-source generalist robot policy. In: Proceedings of Robotics: Science and Systems (2024) 14
56. 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.: Bridgedata v2: A dataset for robot learning at scale. In: Conference on Robot Learning. pp. 1723–1736. PMLR (2023) 2, 15
57. Wang, Y., Ding, P., Li, L., Cui, C., Ge, Z., Tong, X., Song, W., Zhao, H., Zhao, W., Hou, P., Huang, S., Tang, Y., Wang, W., Zhang, R., Liu, J., Wang, D.: Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. In: AAAI (2026) 1, 9, 11
58. Wang, Y., Yue, Z., Zeng, H., Wang, D., McAuley, J.: Train once, deploy anywhere: Matryoshka representation learning for multimodal recommendation. In: Findings of the Association for Computational Linguistics: EMNLP 2024. pp. 13461–13472 (2024) 15
59. Wu, K., Hou, C., Liu, J., Che, Z., Ju, X., Yang, Z., Li, M., Zhao, Y., Xu, Z., Yang, G., et al.: Robomind: Benchmark on multi-embodiment intelligence normative data

for robot manipulation. In: Proceedings of Robotics: Science and Systems (2025) 15
60. Wu, X., Liang, D., Feng, T., Xia, K., Zhang, Y., Li, X., Tan, X., Bai, X.: Generation models know space: Unleashing implicit 3d priors for scene understanding. arXiv preprint arXiv:2603.19235 (2026) 15
61. Xiang, F., Qin, Y., Mo, K., Xia, Y., Zhu, H., Liu, F., Liu, M., Jiang, H., Yuan, Y., Wang, H., Yi, L., Chang, A.X., Guibas, L.J., Su, H.: SAPIEN: A simulated part-based interactive environment. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020) 2, 5
62. Yakefu, A., Xie, B., Xu, C., Zhang, E., Zhou, E., Jia, F., Yang, H., Fan, H., Zhang, H., Peng, H., et al.: Robochallenge: Large-scale real-robot evaluation of embodied policies. arXiv preprint arXiv:2510.17950 (2025) 15
63. Ye, J., Gong, S., Gao, J., Fan, J., Wu, S., Bi, W., Bai, H., Shang, L., Kong, L.: Dream-vl & dream-vla: Open vision-language and vision-language-action models with diffusion language model backbone. arXiv preprint arXiv:2512.22615 (2025) 3
64. Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., Levine, S.: Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In: Conference on robot learning. pp. 1094–1100. PMLR (2020) 15
65. Zhang, J., Wang, K., Xu, R., Zhou, G., Hong, Y., Fang, X., Wu, Q., Zhang, Z., Wang, H.: Navid: Video-based vlm plans the next step for vision-and-language navigation. In: Proceedings of Robotics: Science and Systems (2024) 14
66. Zhang, S., Xu, Z., Liu, P., Yu, X., Li, Y., Gao, Q., Fei, Z., Yin, Z., Wu, Z., Jiang, Y.G., et al.: Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11142–11152 (2025) 15
67. Zhang, W., Liu, H., Qi, Z., Wang, Y., Yu, X., Zhang, J., Dong, R., He, J., Lu, F., Wang, H., et al.: Dreamvla: A vision-language-action model dreamed with comprehensive world knowledge. In: NeurIPS (2025) 3, 7, 15
68. Zhang, Y., Wang, R., Chen, X.: Dynamic behavior cloning with temporal feature prediction: Enhancing robotic arm manipulation in moving object tasks. IEEE Robotics and Automation Letters (2025) 15
69. Zhao, T., Kumar, V., Levine, S., Finn, C.: Learning fine-grained bimanual manipulation with low-cost hardware. Robotics: Science and Systems XIX (2023) 3, 6, 9
70. Zheng, R., Liang, Y., Huang, S., Gao, J., Daumé III, H., Kolobov, A., Huang, F., Yang, J.: Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In: ICLR (2025) 14
71. Zhou, X., Liang, D., Chen, X., Tan, F., Zhang, D., Zhao, H., Bai, X.: Hermes++: Toward a unified driving world model for 3d scene understanding and generation. arXiv preprint arXiv:2604.28196 (2026) 15
72. Zhou, X., Liang, D., Tu, S., Chen, X., Ding, Y., Zhang, D., Tan, F., Zhao, H., Bai, X.: Hermes: A unified self-driving world model for simultaneous 3d scene understanding and generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2025) 15
73. Zhu, Y., Wong, J., Mandlekar, A., Martín-Martín, R., Joshi, A., Lin, K., Maddukuri, A., Nasiriany, S., Zhu, Y.: robosuite: A modular simulation framework and benchmark for robot learning. arXiv preprint arXiv:2009.12293 (2020) 15
74. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al.: Rt-2: Vision-language-action models transfer web

knowledge to robotic control. In: Conference on Robot Learning. pp. 2165–2183. PMLR (2023) 14