WHAT MATTERS FOR LATENT ACTIONS IN ROBOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent Action Models (LAMs) have emerged as a promising paradigm for enabling robot learning to leverage large-scale unlabeled videos through latent actions that serve as compact surrogates for physical actions. Despite rapid progress, research on LAM remains highly fragmented, with existing methods evaluating different design choices in isolation under inconsistent experimental settings, making it difficult to identify the factors that truly determine downstream robotic manipulation performance. In this work, we present the first comprehensive empirical study of latent action learning for robotic manipulation. We unify representative LAM methods within a common autoencoding framework and systematically investigate 41 LAM design choices across three dimensions, including latent action modeling paradigms, learning objectives and regularization methods, and latent action integration strategies. We further examine four proxy metrics for evaluating latent action quality and assess their ability to reliably predict downstream robotic manipulation performance. Extensive experiments on three widely used benchmarks provide strong empirical evidence that fine-tuning vision-language model (VLM) backbones with latent actions provides a stronger initialization for downstream policy learning, with further validation on real-world robot manipulation tasks.

1 INTRODUCTION

Actions drive the transition dynamics of the physical world, yet action modeling fundamentally relies on large-scale robot action data (Kang et al., 2026; Pertsch et al., 2025; Dong et al., 2026). Unfortunately, collecting such data remains prohibitively costly and difficult, limiting the development of scalable action representations. Although embodied foundation models have advanced rapidly in recent years, much of this progress has been inherited from advances in adjacent foundation model paradigms rather than from corresponding advances in action modeling. In particular, vision-language-action (VLA) models (Zitkovich et al., 2023; Black et al., 2024; Kim et al., 2024) have inherited advances in vision-language models (VLMs) (Bai et al., 2025; Beyer et al., 2024; Karamcheti et al., 2024), while world action models (WAMs) (Bi et al., 2026; Ye et al., 2026; Agarwal et al., 2026) have similarly benefited from recent progress in video generation models (VGMs) (Wan et al., 2025; Ali et al., 2025; Zheng et al., 2024). These foundation models are enabled by web-scale image-text and video corpora, whose scale and diversity exceed those of available robot action data by several orders of magnitude. Consequently, current embodied foundation models (Team, 2026; Agarwal et al., 2026) are typically initialized from large-scale pretrained VLMs or VGMs and subsequently adapted to robotic manipulation using comparatively limited robot action data (Ji et al., 2025; Wu et al., 2025; O'Neill et al., 2024). Despite recent efforts have explored leveraging egocentric human videos (Li et al., 2025a; Kareer et al., 2025; Yang et al., 2025) and low-cost teleoperation systems such as universal manipulation interface (UMI) (Chi et al., 2024; Liu et al., 2024; Ohkawa et al., 2026) to scale data collection, the scarcity of large-scale, high-quality robot action data remains unresolved. This imbalance between abundant perceptual data and scarce robot action data fundamentally limits the scalability of embodied foundation models and hinders the emergence of scaling behavior analogous to that observed in the training of large language models (Kaplan et al., 2020).

A natural way to overcome this bottleneck is to exploit large-scale unlabeled videos to train latent action models (LAMs), which extract compact latent actions that serve as surrogates for physical ac-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tions (Edwards et al., 2019; Schmidt & Jiang, 2024; Liu et al., 2025a). A representative framework is LAPO (Schmidt & Jiang, 2024), which learns latent action representations through a self-supervised autoencoding framework consisting of an inverse dynamics model and a forward dynamics model. Specifically, the inverse dynamics model encodes consecutive frames into a low-dimensional latent action that captures the transition dynamics, while the forward dynamics model predicts the future frame from the current frame conditioned on the inferred latent action. This paradigm enables scalable learning from abundant unlabeled video and has demonstrated promising performance in both VLA (Ye et al., 2025; Bu et al., 2025b;a) and WAM (Bi et al., 2026). However, current research on LAM remains fragmented and lacks a coherent empirical understanding. Existing methods vary substantially in how latent action representations are learned from consecutive frames (Bu et al., 2026; Yang et al., 2026; Bi et al., 2026; Liu et al., 2025a), how they are regularized (Bu et al., 2025b; Gao et al., 2025; Maes et al., 2026; Garrido et al., 2026), and how latent actions are integrated into physical action prediction (Gao et al., 2026; Liu et al., 2025b; Lin et al., 2026). Moreover, these design choices are typically proposed in isolation and evaluated under inconsistent experimental settings, making it unclear which factors fundamentally affect latent action learning and which merely have incidental effects. Another major obstacle is the lack of a reliable protocol for evaluating latent action quality. While downstream robotic manipulation performance provides the most faithful evaluation of latent action quality, obtaining such measurements requires completing the full three-stage training pipeline, as illustrated in Fig. 1, making iterative LAM development prohibitively expensive. As a result, prior work has relied on proxy metrics, mostly by training a lightweight multi-layer perceptron (MLP) probe during the pre-training stage to decode latent actions into physical actions and using the prediction error as a proxy for latent action quality (Schmidt & Jiang, 2024; Yang et al., 2026; Bu et al., 2026; Nie et al., 2026). Yet, the validity of this proxy metric remains largely unverified, and whether it reliably predicts downstream robotic manipulation performance remains unclear.

In this work, we conduct a systematic empirical study of LAM design choices across three dimensions, including latent action modeling paradigms, learning objectives and regularization methods, and latent action integration strategies. Specifically, we investigate 41 design choices across the LIBERO, LIBERO-Plus, and RoboTwin2.0 benchmarks, with further validation on real-world robotic manipulation tasks using a Franka Panda robot. We further evaluate four proxy metrics for latent action quality, analyze their correlation with downstream robotic manipulation performance, and investigate the effect of expanding the Stage-II VLM fine-tuning corpus on downstream policy learning. Our contributions are summarized as follows:

(1) The original LAPO method remains a remarkably strong baseline when trained directly on raw data, while simple semantic feature differencing using off-the-shelf visual encoders is also sufficient to yield competitive latent action representations.

(2) We identify effective hyperparameter settings for different regularization methods, showing that properly tuned methods achieve comparable downstream performance. We further evaluate five latent action integration strategies and derive practical guidelines for effectively incorporating latent actions into physical action prediction.

(3) A latent action dimensionality of 32 achieves the highest average performance across the evaluated benchmarks. Additional latent action normalization generally does not improve performance when appropriate pre-training regularization is applied.

(4) FDM reconstruction metrics tend to provide more reliable indications of downstream robotic manipulation success rates than probe metrics. These proxy metrics are better suited to coarse-grained model selection than to fine-grained ranking and are insufficient for reliably identifying the best-performing model.

(5) Both simulation and real-world experiments demonstrate that fine-tuning VLM backbones with latent actions provides stronger initialization for downstream policy learning. Expanding the Stage-II VLM fine-tuning corpus with OXE data improves downstream robotic manipulation performance across the evaluated benchmarks.

2 METHODOLOGY

Training Pipeline Setting The overall framework consists of three stages: pre-training, mid-training, and post-training, corresponding to representation learning, VLM backbone fine-tuning,

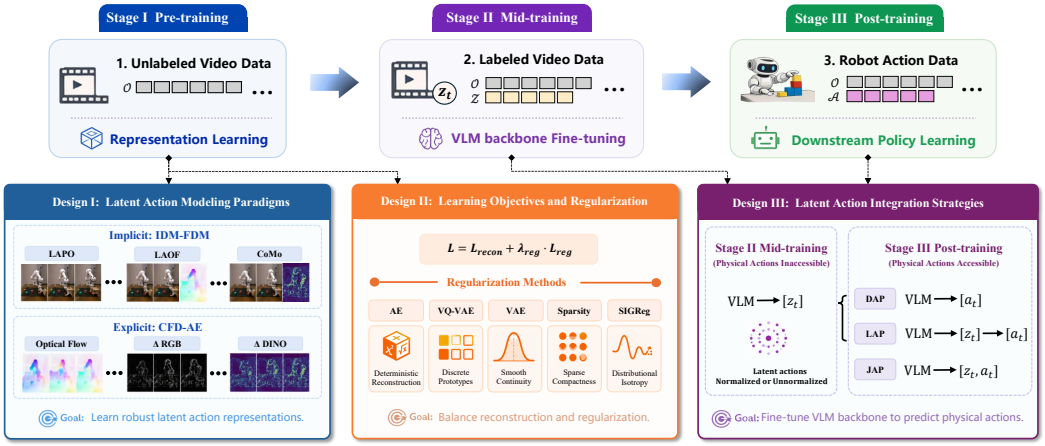

Figure 1: **Overview of design choices across three dimensions for latent action learning.** Design I examines the impact of latent action modeling paradigms by comparing the implicit IDM-FDM framework with the explicit CFD-AE framework. Design II investigates how different regularization methods and their regularization strengths shape latent action representations and affect downstream policy learning. Design III evaluates the necessity of latent action normalization and investigates different strategies for integrating latent actions into physical action prediction. Stage I is trained on unlabeled video data consisting only of image frames, Stage II uses labeled video data paired with latent actions, and Stage III uses robot action data paired with physical actions. Further details of the three design studies and the three-stage training pipeline are provided in Section 2 and Appendix B.2, respectively.

and downstream policy learning, respectively. **Stage I (Pre-training).** This stage aims to learn a high-quality LAM from consecutive frames (o_t, o_{t+1}) to produce physically meaningful latent action representations that capture transition dynamics. **Stage II (Mid-training).** The pretrained inverse dynamics model is employed as a latent action annotator to automatically generate latent actions for semantically conditioned video data, yielding triplets (o_t, z_t, l). These latent actions are then used to fine-tune the VLM backbone, producing the embodied foundation model studied in this work. **Stage III (Post-training).** The fine-tuned VLM backbone is subsequently optimized using a limited amount of robot action data (o_t, a_t, l) for downstream policy learning.

Design I: Latent Action Modeling Paradigms We consider a set of representative methods for latent action modeling, which can be broadly divided into two paradigms, as illustrated in Fig. 8. The first is the inverse dynamics model–forward dynamics model (IDM-FDM) framework, which implicitly learns latent action representations through jointly trained inverse and forward dynamics models, including LAPO (Schmidt & Jiang, 2024) and its variants, LAOF (Bu et al., 2026) and CoMo (Yang et al., 2026). The second is the consecutive-frame differences autoencoding (CFD-AE) framework, which explicitly learns latent action representations by autoencoding consecutive-frame differences $\Delta_t = \Delta(o_{t+1}, o_t)$. In this work, Δ_t is instantiated as ΔRGB, ΔDINO, or optical flow.

Design II: Learning Objectives and Regularization A systematic investigation of latent action learning requires understanding how different learning objectives and regularization methods shape latent action representations, as existing methods lead to substantial differences in the structure, compactness, and controllability of the learned representations. To provide a unified analysis, we consider several representative learning objectives and regularization methods, including AE (Bank et al., 2023), VAE (Kingma & Welling, 2013), VQ-VAE (Van Den Oord et al., 2017), Sparsity (Garrido et al., 2026), and SIGReg (Balestriero & LeCun, 2025).

Design III: Latent Action Integration Strategies We consider three representative latent action integration strategies, namely direct action prediction (DAP), latent-to-action prediction (LAP), and joint latent-action prediction (JAP), which differ in whether latent actions are used only to fine-tune the VLM backbone, explicitly serve as an intermediate control representation, or are jointly

Table 1: Performance comparison across LIBERO, LIBERO-Plus, and RoboTwin2.0. Entries report success rates. Avg. averages over the five integration strategies; red boldface marks the highest Avg. within each design group and benchmark.

Method	LIBERO						LIBERO-Plus						RoboTwin2.0					
	DAP	LAP	JAP	JAP-DAP	JAP-LAP	Avg.	DAP	LAP	JAP	JAP-DAP	JAP-LAP	Avg.	DAP	LAP	JAP	JAP-DAP	JAP-LAP	Avg.
Design I: Latent Action Modeling Paradigms																		
RAFT	0.676	0.714	0.838	0.836	0.837	0.780	0.310	0.359	0.340	0.346	0.356	0.342	0.785	0.773	0.825	0.831	0.815	0.806
SEA-RAFT	0.673	0.764	0.847	0.868	0.858	0.802	0.289	0.339	0.336	0.315	0.351	0.326	0.787	0.804	0.769	0.820	0.833	0.802
ΔRGB	0.853	0.860	0.899	0.894	0.896	0.880	0.352	0.382	0.397	0.388	0.413	0.387	0.795	0.809	0.834	0.856	0.830	0.825
ΔDINO	0.933	0.923	0.913	0.920	0.918	**0.921**	0.458	0.488	0.396	0.408	0.402	0.430	0.855	0.846	0.817	0.813	0.837	0.833
CoMo	0.907	0.912	0.910	0.903	0.908	0.908	0.372	0.399	0.405	0.414	0.396	0.397	0.811	0.862	0.858	0.838	0.858	0.845
LAOF	0.889	0.874	0.895	0.889	0.892	0.888	0.325	0.285	0.387	0.377	0.395	0.354	0.848	0.810	0.812	0.843	0.836	0.830
LAPO	0.928	0.919	0.907	0.898	0.900	0.910	0.397	0.430	0.449	0.452	0.455	**0.437**	0.862	0.845	0.830	0.862	0.856	**0.851**
Design II: Learning Objectives and Regularization																		
VAE(1e-7)	0.921	0.913	0.905	0.904	0.906	0.910	0.452	0.463	0.400	0.424	0.425	0.433	0.838	0.875	0.860	0.870	0.852	**0.859**
Sparsity(1e-5)	0.920	0.915	0.913	0.915	0.916	**0.915**	0.471	0.501	0.459	0.451	0.461	0.468	0.822	0.844	0.795	0.848	0.846	0.831
SIGReg(1e-3)	0.910	0.909	0.902	0.909	0.913	0.908	0.481	0.478	0.462	0.442	0.469	0.467	0.817	0.847	0.829	0.865	0.849	0.841
AE	0.928	0.925	0.901	0.906	0.904	0.913	0.440	0.455	0.436	0.449	0.447	0.445	0.828	0.844	0.859	0.873	0.856	0.852
VQ-VAE(1)	0.920	0.910	0.905	0.907	0.909	0.910	0.537	0.534	0.494	0.498	0.522	**0.517**	0.811	0.799	0.830	0.821	0.845	0.821

predicted with physical actions. To evaluate whether a jointly fine-tuned backbone can be used with different downstream action heads, we additionally design JAP-DAP and JAP-LAP. Specifically, in Stage III, the VLM backbone is first jointly fine-tuned with both latent actions and physical actions following the JAP setting. The trained action head is then discarded, and a new DAP or LAP head is trained from scratch.

3 Experiments

Our experiments are organized into three dimensions, each comprising multiple design choices that examine different aspects of latent action learning. Design I focuses on latent action modeling paradigms with 7 choices. Design II covers learning objectives and regularization, with 21 choices spanning different regularization methods and their corresponding strengths. Design III studies latent action integration, with 15 choices covering latent action normalization, dimensionality, and integration strategies. After removing overlapping configurations, the three dimensions yield 41 unique design choices in total. Additional results and analyses are provided in Appendix C.

3.1 Experimental Setup

Datasets. In Stage I and Stage II, we use only raw video data without accessing robot action annotations. To investigate the effect of expanding the Stage-II VLM fine-tuning corpus, we construct a heterogeneous video-only corpus containing approximately 59M frames, whose composition is illustrated in Fig. 6. Unless otherwise specified, all experiments in Stages I and II use the full data mixture, while large-scale ablation studies of Design I–III are conducted on the `Robotwin` and `liberoplus` datasets.

Implementation Details. In Stage I, we implement LAM as a 700M spatiotemporal Transformer (Bruce et al., 2024). In Stages II and III, we adopt OpenVLA-OFT (Kim et al., 2025) as the baseline following the implementation of StarVLA (Community, 2026), and use Qwen3-VL-4B (Bai et al., 2025) as the VLM backbone. To eliminate confounding factors when evaluating the effect of the VLM fine-tuned with latent actions on downstream robotic manipulation performance in Stage III, both training and evaluation use only front-view observations, without robot joint states, wrist-view observations, or any data augmentation. All experiments are conducted on 8 NVIDIA H200 GPUs.

Benchmark. Our simulation experiments cover two control settings: 7-DoF single-arm end-effector delta control and 14-DoF dual-arm joint position control. Unless otherwise specified, all ablation results are reported as the average success rate across 3 random seeds. Specifically, LIBERO results are averaged over its 4 task suites, LIBERO-Plus results are averaged over the same 4 task suites under the `RobotInit` perturbation, and RoboTwin2.0 results are averaged over 12 selected tasks, with each task evaluated under both easy and hard modes. Here, zero-shot evaluation on LIBERO-Plus refers to deployment under `RobotInit` after Stage-III policy training on the `Libero` datasets only; `liberoplus` videos are used in Stages I and II.

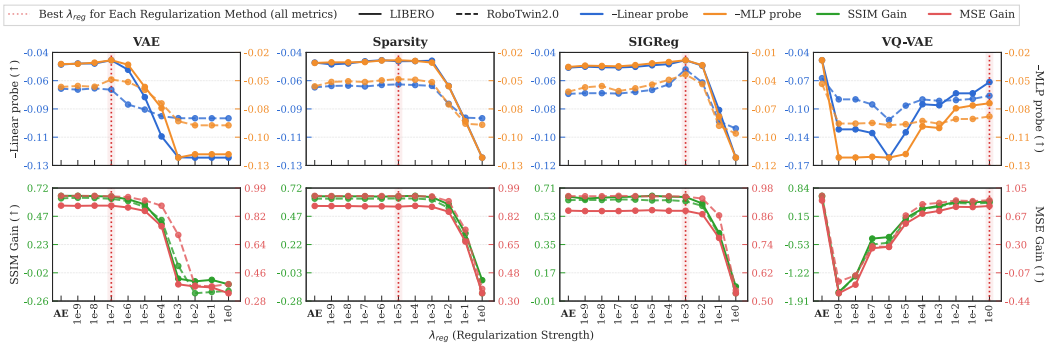

Figure 2: **Effect of regularization strength λ_{reg} on latent action learning.** Columns show four regularization methods, with AE as the unregularized baseline. The top and bottom rows show probe and reconstruction metrics (Section 3.5), respectively, on color-matched vertical axes. Solid and dashed curves denote LIBERO and RoboTwin2.0; dotted vertical lines mark selected strengths.

3.2 Which latent action modeling paradigms produce the highest-quality latent action representations? (**Q1**)

As shown in Table 1, we compare seven choices of latent action modeling paradigms across the LIBERO, LIBERO-Plus, and RoboTwin2.0 benchmarks, with each choice evaluated using five latent action integration strategies, including DAP, LAP, JAP, JAP-DAP, and JAP-LAP. Here, *Avg.* denotes the average performance across the five latent action integration strategies. All methods are evaluated with latent action dimensionality $d_z = 64$ and regularization strength $\lambda_{reg} = 1 \times 10^{-6}$. Averaging over all benchmarks, the overall ranking is LAPO (0.7327) > ΔDINO (0.7280) > CoMo (0.7167) > ΔRGB (0.6973) > LAOF (0.6907) > SEA-RAFT (0.6433) > RAFT (0.6427). LAPO achieves the best overall performance, suggesting that it learns more robust latent action representations for robotic manipulation. Notably, LAPO requires no additional preprocessing and can be trained directly on raw data, highlighting its simplicity and effectiveness.

With optical flow computed directly from the raw data and without any additional filtering, SEA-RAFT and RAFT consistently achieve the lowest performance across all three benchmarks, even underperforming the simple ΔRGB baseline. Moreover, LAOF extends LAPO by incorporating optical flow as an additional constraint, yet this modification consistently degrades performance.

Furthermore, ΔDINO consistently outperforms ΔRGB across all benchmarks. This result suggests that the representation space in which consecutive-frame differences are computed plays an important role in latent action modeling. Notably, ΔDINO achieves overall performance close to that of LAPO and even outperforms LAPO on LIBERO, suggesting that a strong pretrained semantic prior can itself provide an effective inductive bias for latent action learning in robotic manipulation. Finally, LAPO consistently outperforms CoMo across all three benchmarks. The primary difference between the two methods is that CoMo replaces the future-frame input to the IDM with semantic feature differences to mitigate potential information leakage, but this modification consistently degrades downstream robotic manipulation performance.

3.3 How do regularization methods and their regularization strengths affect latent action learning? (**Q2**)

We find that the choice of regularization type has a relatively limited impact on downstream robotic manipulation performance, whereas the regularization strength plays a much more critical role in determining latent action quality, as shown in Fig. 2. Unless otherwise specified, all Design-II results are obtained using LAPO with a latent action dimensionality of $d_z = 64$.

For continuous latent actions, VAE, Sparsity, and SIGReg achieve comparable downstream performance on both LIBERO and RoboTwin2.0, as shown by the Design-II results in Table 1. For example, on LIBERO, their average success rates range only from 0.908 to 0.915. VQ-VAE achieves the strongest zero-shot generalization on LIBERO-Plus, with an average success rate of 0.517, outper-

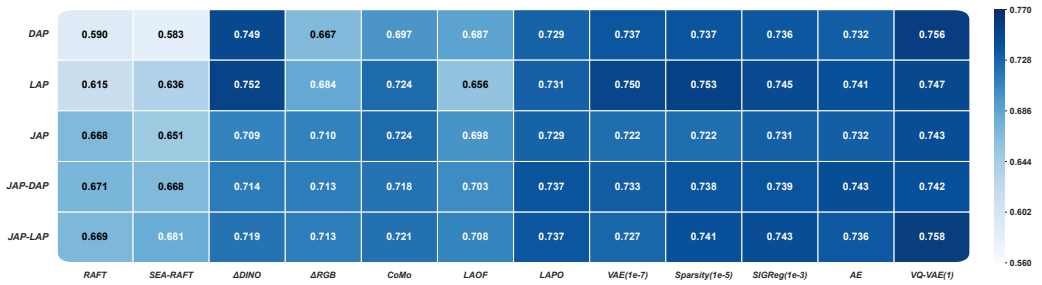

Figure 3: **Performance comparison of different latent action integration strategies.** The horizontal axis shows seven latent action modeling paradigms and five regularization methods. The vertical axis shows five latent action integration strategies. Each cell reports the mean success rate across LIBERO, LIBERO-Plus, and RoboTwin2.0.

forming AE (0.445), VAE (0.433), Sparsity (0.468), and SIGReg (0.467). This observation suggests that discretizing latent actions may improve robustness under distribution shifts.

Weak-to-moderate regularization is generally the safest choice for continuous latent actions, and we recommend $\lambda_{\text{reg}} = 10^{-7}$ for VAE, 10^{-5} for Sparsity, and 10^{-3} for SIGReg. For discrete latent actions, $\lambda_{\text{reg}} = 1$ is recommended for VQ-VAE. VAE is thus the recommended default when simplicity and effectiveness are prioritized, while VQ-VAE is preferred when zero-shot generalization is the primary concern.

3.4 How should latent actions be integrated into physical action prediction? (**Q3**)

We consider two settings for incorporating latent actions into downstream policy learning: Physical Action Inaccessible and Physical Action Accessible. The distinction is whether physical action annotations are available during VLM backbone fine-tuning with latent actions. DAP and LAP are evaluated in the former setting, and JAP, JAP-DAP, and JAP-LAP in the latter. Both settings use robot action annotations for downstream policy learning.

Physical Action Inaccessible. In this setting, we compare DAP and LAP, both of which use only latent actions to fine-tune the VLM backbone in Stage II. Unlike DAP, LAP preserves latent actions as an intermediate representation between VLM features and physical actions. As shown in Fig. 3, LAP outperforms DAP in 10 of the 12 configurations after averaging over the three benchmarks for each configuration. Equally weighting the 12 configurations and three benchmarks in Table 1, LAP achieves a higher overall mean success rate than DAP (0.7114 vs. 0.7000). In addition, its mean is close to that of JAP (0.7115), which uses physical action annotations during joint fine-tuning. Therefore, we recommend LAP when only video data are available for Stage II.

Physical Action Accessible. In this setting, we compare JAP and its two variants, JAP-DAP and JAP-LAP, all of which jointly use latent actions and physical actions to fine-tune the VLM backbone. JAP jointly predicts latent and physical actions from shared VLM features. To evaluate whether a jointly fine-tuned backbone can be used with different downstream action heads, we introduce JAP-DAP and JAP-LAP. Both variants retain the jointly fine-tuned backbone, discard the joint prediction head, and train a new DAP or LAP head from scratch. Across the same 12 configurations and three benchmarks, JAP-LAP achieves the highest overall mean success rate among these three strategies (0.7212), followed by JAP-DAP (0.7181) and JAP (0.7115). When robot action annotations are available for joint fine-tuning, JAP-LAP is the recommended choice based on its overall mean performance; the best strategy for an individual configuration or benchmark can differ.

3.5 Are proxy metrics for latent actions predictive of downstream robotic manipulation performance? (**Q4**)

Proxy Metrics. We evaluate latent action quality using four proxy metrics and analyze their correlation with downstream robotic manipulation performance. Linear probe and MLP probe evaluate the information about physical actions preserved in latent actions by measuring the accuracy of

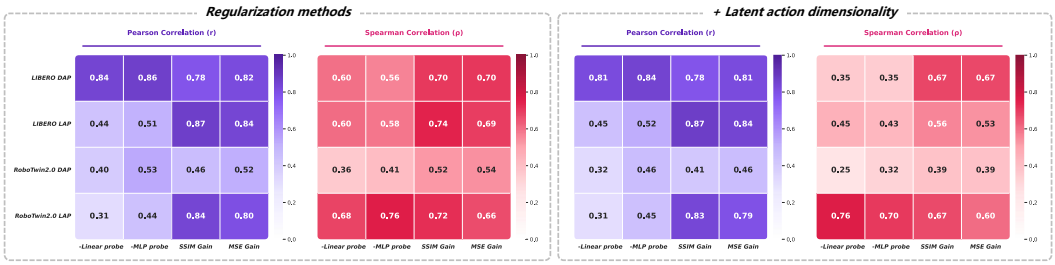

Figure 4: **Correlation between proxy metrics and downstream robotic manipulation performance.** Within each panel, columns correspond to the four proxy metrics and rows to benchmark–strategy combinations (LIBERO/RoboTwin2.0 with DAP/LAP). The purple and red panels report the Pearson correlation (r) and Spearman correlation (ρ), respectively.

physical action recovery using lightweight action decoders. Specifically, the Linear probe employs a linear action decoder with architecture $[d_z, d_a]$, whereas the MLP probe adopts a nonlinear action decoder with architecture $[d_z, 128, 128, d_a]$. To ensure a fair comparison across different latent action representations, both probes are trained under the same settings with a learning rate of 1×10^{-3}, 50 training steps, and a global batch size of 256.

Lower $\mathcal{L}_{\text{act}}$, the physical action prediction loss, indicates that latent actions preserve more information relevant to physical action prediction. Accordingly, we report $-$Linear probe and $-$MLP probe in the correlation analysis so that larger values consistently indicate higher latent action quality across all proxy metrics. SSIM Gain and MSE Gain are normalized reconstruction metrics that quantify the improvement of future-frame reconstruction over directly using the current frame as the prediction. Specifically, SSIM Gain is defined as $[\text{SSIM}(o_{t+1}, \hat{o}_{t+1}) - \text{SSIM}(o_t, o_{t+1})]/[1 - \text{SSIM}(o_t, o_{t+1})]$, where SSIM denotes the structural similarity index and $\hat{o}_{t+1}$ denotes the FDM-predicted future frame. MSE Gain is defined as $[\text{MSE}(o_t, o_{t+1}) - \text{MSE}(o_{t+1}, \hat{o}_{t+1})]/\text{MSE}(o_t, o_{t+1})$. Larger SSIM Gain and MSE Gain indicate that the learned latent actions provide more informative transition dynamics, enabling the FDM to transform the current frame into the future frame rather than simply copying the current frame.

Our correlation analysis suggests that the proposed proxy metrics provide useful signals of downstream robotic manipulation performance, but they should not be regarded as a complete replacement for task evaluation. As shown in Fig. 4, the Pearson and Spearman correlations are positive but vary across the evaluated settings, and proxy scores do not fully preserve the downstream ordering of configurations, indicating that the proxy metrics are more reliable for coarse-grained model screening than for fine-grained model ranking. In other words, the proxy metrics can help identify promising configurations, but downstream task evaluation remains necessary for model selection.

Overall, FDM reconstruction metrics (SSIM Gain and MSE Gain) tend to reflect downstream robotic manipulation success rates more consistently than probe metrics ($-$Linear probe and $-$MLP probe). Reconstruction metrics assess future-frame prediction, whereas probe metrics assess physical action recovery through an additional decoder. We also find that the correlations of all proxy metrics deteriorate when comparing models with different latent action dimensionalities. We therefore recommend using these proxy metrics to compare models with the same latent action dimensionality rather than across different dimensionalities. As shown in Fig. 9, it can be observed that LIBERO under end-effector delta control consistently yields lower probe losses than RoboTwin2.0 under joint position control over the training steps shown.

3.6 What is the most suitable latent action dimensionality, and is latent action normalization necessary? (**Q5**)

For both 7-DoF single-arm and 14-DoF bimanual robot control, a latent action dimensionality of $d_z = 32$ provides the best overall trade-off, as shown in Fig. 5. On LIBERO, a 7-DoF single-arm benchmark, the downstream robotic manipulation performance is largely saturated with respect to the latent dimensionality. Although the highest average success rate is achieved with $d_z = 16$ (0.926), the performance of $d_z = 32$ is nearly identical (0.922). A similar trend is observed on

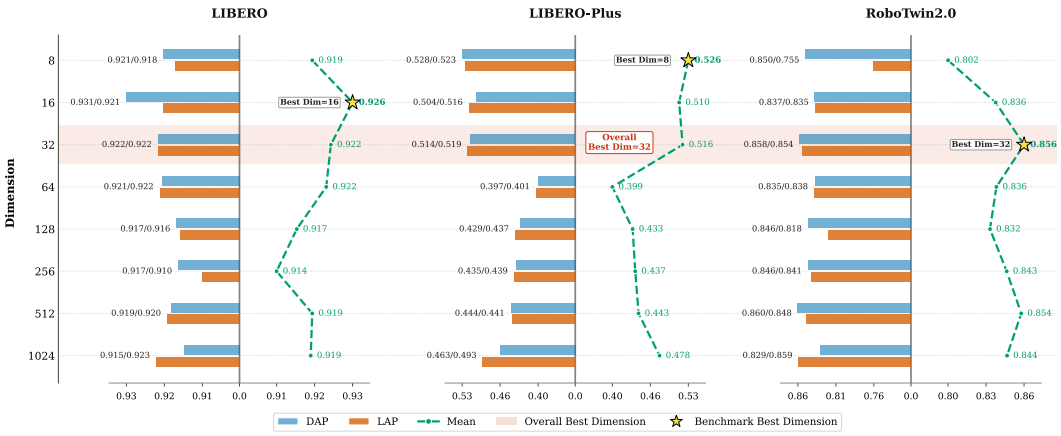

Figure 5: **Effect of latent action dimension on downstream robotic manipulation performance.** Blue (DAP) and orange (LAP) bars show success rates across latent action dimensions; the green dashed curve shows their mean. Stars mark the best dimension per benchmark; shading marks the best dimension averaged across benchmarks. Results use LAPO with VAE regularization (10^{-6}).

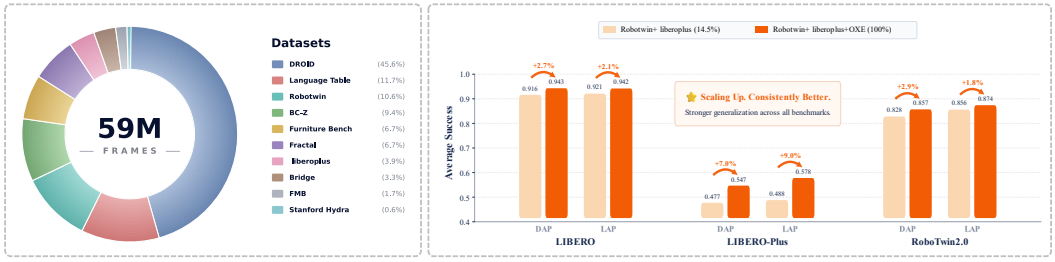

Figure 6: **Effect of Stage-II corpus expansion on downstream robotic manipulation performance.** The left panel shows the full corpus composition; `OXE` includes all datasets except `Robotwin` and `liberoplus`. The right panel reports average DAP and LAP success rates by benchmark. Light and dark orange bars denote Stage-II training without and with `OXE`, respectively.

LIBERO-Plus, which also evaluates 7-DoF single-arm manipulation while introducing diverse environmental perturbations to better assess policy robustness. More compact latent representations perform slightly better, with $d_z = 8$ achieving the highest average success rate of 0.526, whereas $d_z = 32$ remains highly competitive at 0.516. In contrast, RoboTwin2.0 evaluates 14-DoF bimanual manipulation, which requires coordinated dual-arm motions and therefore places higher demands on representation capacity. Under this setting, $d_z = 32$ achieves the best overall performance (0.856), substantially outperforming $d_z = 8$ (0.802).

We further investigate the effect of normalizing latent actions for VLM backbone fine-tuning in Stage II. As shown in Fig. 10, when the latent space is appropriately regularized, using unnormalized latent actions generally yields better downstream robotic manipulation performance. Across 33 method and benchmark combinations, 28 exhibit performance improvements without normalization, with an average gain of $+0.0115$. The largest improvement is observed on LIBERO-Plus ($+0.0160$), indicating that preserving the original scale of latent actions is particularly beneficial under challenging distribution shifts.

3.7 Does fine-tuning the VLM backbone with latent actions exhibit scaling laws in robotic manipulation? (**Q6**)

Expanding the Stage-II fine-tuning corpus with OXE data improves downstream robotic manipulation performance while keeping Stage III unchanged. As shown in Fig. 6, expanding the Stage-II fine-tuning corpus with OXE data improves performance across the evaluated benchmarks. The

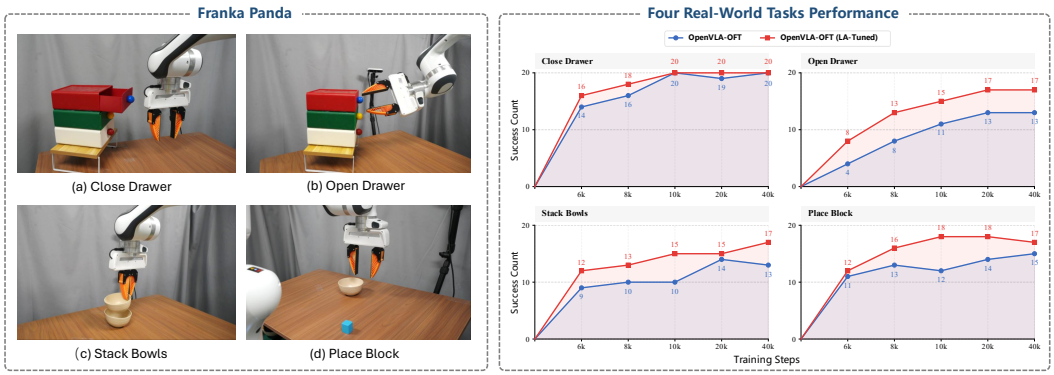

Figure 7: **Real-World Experiments.** The left panel shows the four tasks on the Franka Panda platform. The right panel plots success counts out of 20 trials against Stage-III training steps. Blue circles and red squares denote OpenVLA-OFT and OpenVLA-OFT (LA-Tuned), respectively.

largest improvement is observed on LIBERO-Plus, with gains of up to 9.0 percentage points, suggesting that expanding the fine-tuning corpus is particularly beneficial for this benchmark. In comparison, the improvements on LIBERO and RoboTwin2.0 are smaller but remain consistently positive, indicating that the benefit of this corpus expansion extends across the evaluated benchmarks. Moreover, both DAP and LAP consistently benefit from the expanded Stage-II fine-tuning corpus across the evaluated benchmarks.

Real-World Experiments. To further evaluate the effectiveness of fine-tuning the VLM backbone using latent actions in real-world environments, we conduct experiments on a 7-DoF single-arm Franka Panda platform equipped with a 1-DoF UMI gripper, as illustrated in the left panel of Fig. 7. Further experimental details are provided in Appendix C.8.

As shown in the right panel of Fig. 7, fine-tuning the VLM backbone with latent actions consistently improves real-world manipulation performance on the Franka Panda platform. Across all four tasks and five Stage III checkpoints, OpenVLA-OFT (LA-Tuned) achieves 317 successful rollouts out of 400, compared with 259/400 for the original OpenVLA-OFT baseline, improving the overall success rate from 64.75% to 79.25% (+14.5 percentage points, or +22.4% relative improvement). The gains are observed on all task categories, including stacking, object placement, and articulated-object manipulation. Notably, the LA-Tuned backbone reaches a higher success rate earlier in Stage III training, achieving an average success rate of 85.0% after only 10k Stage III training steps, already exceeding the baseline performance of 76.25% obtained after 40k steps. These results indicate that fine-tuning the VLM backbone with latent actions provides a stronger initialization for understanding physical-world interactions, leading to faster Stage III learning and stronger real-world robotic manipulation performance.

4 Conclusion and Future Work

This work presents a systematic empirical study of latent action modeling paradigms, regularization methods, and integration strategies. LAPO remains a strong baseline, and LAP is a practical choice when only video data are available for Stage II. Proxy metrics are better suited to coarse-grained model screening than to fine-grained ranking. Real-world experiments further demonstrate the benefits of latent-action tuning for downstream policy learning.

Future work will explore incorporating latent actions into the pre-training of vision-language or multimodal foundation models. Jointly learning semantic and action representations from scratch may enable stronger alignment between language semantics and physical actions. We also plan to leverage large-scale in-the-wild video data from online platforms, such as YouTube, to expand the diversity and scale of training data. Finally, we will investigate whether our findings generalize across a broader range of robotic platforms, including dexterous hands, quadruped robots, and humanoid robots.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript polishing, including language refinement, structural editing, and checks of terminology and consistency in the presentation and interpretation of the reported results. The authors take full responsibility for the final content of this paper.

REPRODUCIBILITY STATEMENT

Appendix B provides model definitions, objectives, and action integration strategies. Section 3.5 defines the proxy metrics. Appendix C documents the datasets, preprocessing, evaluation protocols, and training hyperparameters.

REFERENCES

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders. *Machine learning for data science handbook: data mining and knowledge discovery handbook*, pp. 353–374, 2023.

Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Hydra: Hybrid robot actions for imitation learning. *arxiv*, 2023.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2026.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *ICML*, 2024.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *IROS*, 2025a.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025b.

Xizhou Bu, Jiexi Lyu, Fulei Sun, Ruichen Yang, Zhiqiang Ma, and Wei Li. Laof: Robust latent action learning with optical flow constraints. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 27334–27344, 2026.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, 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*, 2025a.

Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. Igor: Image-goal representations are the atomic control units for foundation models in embodied ai. *arXiv preprint arXiv:2411.00785*, 2024.

Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, et al. Villa-x: enhancing latent action modeling in vision-language-action models. In *Int. Conf. Learn. Represent.*, 2026.

Yi Chen, Yuying Ge, Weiliang Tang, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for learning robot manipulation from videos. In *Int. Conf. Comput. Vis.*, 2025b.

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *RSS*, 2024.

Jeremy A Collins, Loránd Cheng, Kunal Aneja, Albert Wilcox, Benjamin Joffe, and Animesh Garg. Amplify: Actionless motion priors for robot learning from videos. *arXiv preprint arXiv:2506.14198*, 2025.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Harald Cramér and Herman Wold. Some theorems on distribution functions. *Journal of the London Mathematical Society*, 1(4):290–294, 1936.

Zichen Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. Dynamo: In-domain dynamics pretraining for visuo-motor control. In *Adv. Neural Inform. Process. Syst.*, 2024.

Zibin Dong, Yicheng Liu, Shiduo Zhang, Baijun Ye, Yifu Yuan, Fei Ni, Jingjing Gong, Xipeng Qiu, Hang Zhao, Yinchuan Li, et al. Actioncodec: What makes for good action tokenizers. *arXiv preprint arXiv:2602.15397*, 2026.

Ashley Edwards, Himanshu Sahni, Yannick Schroecker, and Charles Isbell. Imitating latent policies from observation. In *ICML*, 2019.

Thomas W Epps and Lawrence B Pulley. A test for normality based on the empirical characteristic function. *Biometrika*, 70(3):723–726, 1983.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. In *ICML*, 2025.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild. *arXiv preprint arXiv:2601.05230*, 2026.

Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems*, 2023.

Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-z: Zero-shot task generalization with robotic imitation learning. In *CoRL*, 2021.

Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 1724–1734, 2025.

Xirui Kang, Yanpei Shi, Lucy Liang, Roy Gan, Dongxiu Liu, Pushi Zhang, Danpeng Chen, Xiaoyi Qin, Yinan Zheng, Jinliang Zheng, et al. X-tokenizer: A multimodal action tokenizer for vision-language-action pretraining. *arXiv preprint arXiv:2606.14752*, 2026.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. doi: 10.48550/arXiv.2001.08361. URL https://arxiv.org/abs/2001.08361.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *ICML*, 2024.

Simar Kareer, Karl Pertsch, James Darpinian, Judy Hoffman, Danfei Xu, Sergey Levine, Chelsea Finn, and Suraj Nair. Emergence of human to robot transfer in vision-language-action models. *arXiv preprint arXiv:2512.22414*, 2025.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Qixiu Li, Yu Deng, Yaobo Liang, Lin Luo, Lei Zhou, Chengtang Yao, Lingqi Zeng, Zhiyuan Feng, Huizhi Liang, Sicheng Xu, et al. Scalable vision-language-action model pretraining for robotic manipulation with real-life human activity videos. *arXiv preprint arXiv:2510.21571*, 2025a.

Zuolei Li, Xingyu Gao, Xiaofan Wang, and Jianlong Fu. Latbot: Distilling universal latent actions for vision-language-action models. *arXiv preprint arXiv:2511.23034*, 2025b.

Anthony Liang, Pavel Czempin, Matthew Hong, Yutai Zhou, Erdem Biyik, and Stephen Tu. Clam: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025.

Yihan Lin, Haoyang Li, Yang Li, Haitao Shen, Yihan Zhao, Chao Shao, and Jing Zhang. From pixels to tokens: A systematic study of latent action supervision for vision-language-action models. *arXiv preprint arXiv:2605.04678*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *Adv. Neural Inform. Process. Syst.*, 2023.

Kehui Liu, Chuyue Guan, Zhongjie Jia, Ziniu Wu, Xin Liu, Tianyu Wang, Shuai Liang, Pengan Chen, Pingrui Zhang, Haoming Song, et al. Fastumi: A scalable and hardware-independent universal manipulation interface with dataset. *arXiv preprint arXiv:2409.19499*, 2024.

Mingyu Liu, Jiuhe Shu, Hui Chen, Zeju Li, Canyu Zhao, Jiange Yang, Shenyuan Gao, Hao Chen, and Chunhua Shen. Stamo: Unsupervised learning of generalizable robot motion from compact state representation. *arXiv preprint arXiv:2510.05057*, 2025a.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *Int. Conf. Learn. Represent.*, volume 2025, pp. 29982–30009, 2025b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. Fmb: a functional manipulation benchmark for generalizable robotic learning. *arXiv preprint arXiv:2401.08553*, 2024.

Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

Jiexi Lyu, Xizhou Bu, Qingqiu Huang, Chufeng Tang, Xiaoshuai Hao, Hongbo Wang, and Wei Li. Lafp: Preserving latent action structure in latent policy learning via flow matching. *arXiv preprint arXiv:2606.10517*, 2026.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Dujun Nie, Fengjiao Chen, Qi Lv, Jun Kuang, Xiaoyu Li, Xuezhi Cao, and Xunliang Cai. Lary: A latent action representation yielding benchmark for generalizable vision-to-action alignment. *arXiv preprint arXiv:2604.11689*, 2026.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. In *ICML*, 2025.

Takehiko Ohkawa, Jumpei Arima, Yuki Noguchi, Masatoshi Tateno, Makoto Sugiura, Takuya Okubo, Kengo Ikeuchi, Yuma Shin, Hiroki Nishizawa, Naoaki Kanazawa, et al. Yubi: Yielding universal bidigital interface for bimanual dexterous manipulation at scale. *arXiv preprint arXiv:2606.10244*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *TMLR*, 2024. ISSN 2835-8856.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration. In *ICRA*, 2024.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *Int. Conf. Learn. Represent.*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/27985d21f0b751b933d675930aa25022-Abstract-Conference.html.

Generalist Team. Gen-1: Scaling embodied foundation models to mastery. *Generalist AI Blog*, 2026. https://generalistai.com/blog/gen-1.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Eur. Conf. Comput. Vis.*, 2020.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Adv. Neural Inform. Process. Syst.*, volume 30, 2017.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *CoRL*, 2023.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Tianyou Wang, Anson Lei, Joe Watson, and Ingmar Posner. Imitation from heterogeneous demonstrations using grounded latent-action world models. *arXiv preprint arXiv:2606.21672*, 2026.

Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *Eur. Conf. Comput. Vis.*, 2024.

Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, Bowen Yang, Zhe Li, Kai Zhu, Hongyu Wu, Yiheng Liu, et al. Robocoin: An open-sourced bimanual robotic data collection for integrated manipulation. *arXiv preprint arXiv:2511.17441*, 2025.

Jiange Yang, Yansong Shi, Haoyi Zhu, Mingyu Liu, Kaijing Ma, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Como: Learning continuous latent motion from internet videos for scalable robot learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 42352–42363, 2026.

Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Xuxin Cheng, Ri-Zhao Qiu, et al. Egovla: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *Int. Conf. Learn. Represent.*, volume 2025, pp. 28213–28239, 2025.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Chubin Zhang, Jianan Wang, Zifeng Gao, Yue Su, Tianru Dai, Cai Zhou, Jiwen Lu, and Yansong Tang. Clap: Contrastive latent action pretraining for learning vision-language-action models from human videos. *arXiv preprint arXiv:2601.04061*, 2026.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, pp. 2165–2183. PMLR, 2023.

A Related Work

A.1 Latent Action Learning from Consecutive Frames

Different methods exploit different sources of motion signals, ranging from semantic feature differences (Liu et al., 2025a) and optical flow (Bi et al., 2026) to implicitly learned transition dynamics (Schmidt & Jiang, 2024; Bu et al., 2026; Yang et al., 2026). However, it remains unclear which method is most effective for learning high-quality latent action representations that support downstream policy learning. Existing methods can be broadly categorized into two paradigms. The first paradigm models implicit transition dynamics through jointly learned inverse and forward dynamics models (IDM-FDM) under a self-supervised autoencoding framework (Schmidt & Jiang, 2024). Building on this, LAOF (Bu et al., 2026) introduces optical flow as an additional supervision signal during latent action learning to improve robustness to environmental distractors, while CoMo (Yang et al., 2026) mitigates shortcut learning, where the IDM directly encodes future-frame information, by replacing the future-frame input with the difference between consecutive frames. Several subsequent methods (Chen et al., 2026; Liang et al., 2025; Li et al., 2025b; Wang et al., 2026) further incorporate physical actions or robot states to inject robot control priors into latent action representations, leading to improved downstream robotic manipulation performance. Nevertheless, existing robot datasets remain highly heterogeneous, as different robot platforms employ different control interfaces, action space definitions, and control dimensionalities. To avoid these embodiment-specific confounding factors and focus on the intrinsic properties of latent action learning, we adopt a video-only learning setting throughout this stage. The second paradigm, which we refer to as consecutive-frame differences autoencoding (CFD-AE), explicitly learns latent action representations using predefined motion signals. For example, StaMo (Liu et al., 2025a) computes semantic feature differences using an off-the-shelf visual encoder such as DINOv2 (Oquab et al., 2024), whereas Motus (Bi et al., 2026) directly extracts dense optical flow. To enable a fair comparison with implicit IDM-FDM methods, we uniformly transform both semantic feature differences and optical flow into latent actions through a unified autoencoding pipeline, thereby aligning the latent action dimensionality while eliminating architectural differences across methods.

A.2 Regularization in Latent Action Learning

In the standard IDM-FDM framework, the IDM infers a latent action from a pair of consecutive frames, while the FDM predicts the future frame conditioned on the current frame and the inferred latent action. This framework introduces a causal leakage issue, as the IDM has direct access to the target future frame during self-supervised training, potentially leading to shortcut learning. As a result, the inferred latent actions may encode excessive future frame information rather than the underlying transition dynamics, or even degenerate into a direct feature of the future frame itself (Yang et al., 2026). A common strategy to mitigate this issue is to introduce an information bottleneck into the latent action representations. Several methods adopt discretization through VQ-VAE (Van Den Oord et al., 2017), which has been widely used in latent action learning (Bruce et al., 2024; Bu et al., 2025b; Ye et al., 2025; Bu et al., 2025a; Bjorck et al., 2025; Chen et al., 2024; 2026; 2025b; Cui et al., 2024; Collins et al., 2025; Zhang et al., 2026; Lin et al., 2026; Lyu et al., 2026). By learning a finite set of discrete prototypes, VQ-VAE effectively constrains the information capacity of latent action representations. This design was originally motivated by relatively simple discrete-control domains, such as pixel-based games (Bruce et al., 2024), where physical actions can be naturally represented as discrete tokens. For robotic manipulation involving complex continuous control, recent works have increasingly adopted continuous latent action representations based on VAE (Kingma & Welling, 2013), which provide greater flexibility and smoother transitions (Nikulin et al., 2025; Gao et al., 2025; 2026; Yang et al., 2026; Bu et al., 2026; Liang et al., 2025; Li et al., 2025b; Bi et al., 2026). In this setting, latent action representations are regularized by matching their distribution to a Gaussian distribution through a Kullback-Leibler (KL) divergence (Kullback & Leibler, 1951) objective. Other regularization objectives have also been proposed to improve the statistical properties of latent action representations and mitigate representation collapse. For example, Sparsity (Garrido et al., 2026) introduces a more sophisticated regularization objective that encourages sparse and compact representations. In contrast, LeWorldModel (Maes et al., 2026) employs the Sketched-Isotropic-Gaussian Regularizer (SIGReg) (Balestriero & LeCun, 2025), a simpler approach that promotes distributional isotropy by encouraging latent action representations to follow an isotropic Gaussian distribution.

B Detailed Methodology

This appendix supplements Section 2 with notation, learning objectives, and training details.

B.1 Problem Setting

We consider a sequential decision-making setting in which, at each time step t, an agent receives a current frame $o_t \in \mathcal{O}$, executes a physical action $a_t \in \mathcal{A}$, and transitions to the future frame o_{t+1}, where $\mathcal{O}$ denotes the observation space and $\mathcal{A} \subseteq \mathbb{R}^{d_a}$ denotes the physical action space of dimension d_a. Latent action learning aims to learn a latent action representation $z_t \in \mathcal{Z}$ from consecutive frames that captures the underlying transition dynamics, where $\mathcal{Z} \subseteq \mathbb{R}^{d_z}$ denotes the latent action space of dimension d_z. To provide a unified formulation, we denote the consecutive-frame differences by $\Delta_t = \Delta(o_{t+1}, o_t)$, where $\Delta(\cdot, \cdot)$ denotes a function that extracts differences from consecutive frames. Under this formulation, a trajectory is represented as $\tau = (o_t, a_t, z_t, o_{t+1}, l)_{t=1}^{T}$, where l denotes the language instruction associated with the trajectory.

B.2 Training Pipeline Setting

The three-stage pipeline follows Section 2. Stage I requires neither robot action data nor task-specific policy demonstrations, and can be trained entirely on large-scale video data, including videos collected through random exploration or from the web. The stage-specific objectives and action-head initialization protocols are detailed below.

B.3 Latent Action Modeling Paradigms

ΔRGB and ΔDINO are computed as frame differences in pixel space and in the semantic feature space of DINOv2 (Oquab et al., 2024), respectively, while optical flow is estimated using RAFT (Teed & Deng, 2020) or SEA-RAFT (Wang et al., 2024). This setup allows us to assess whether semantic feature differences are more informative than raw pixel differences, and further examine the impact of optical flow quality. All methods in Fig. 8 share the same autoencoding pipeline, ensuring that they are evaluated under the same latent action dimensionality for a fair comparison.

a) LAPO (Schmidt & Jiang, 2024): It is the standard self-supervised IDM-FDM framework for latent action learning. Given consecutive frames (o_t, o_{t+1}), the IDM infers a latent action z_t, and the FDM predicts the future frame conditioned on the current frame and the inferred latent action:

$$z_t = \mathrm{IDM}(o_t, o_{t+1}), \quad \hat{o}_{t+1} = \mathrm{FDM}(o_t, z_t). \tag{1}$$

The two models are jointly optimized in a self-supervised manner using the future-frame reconstruction objective

$$\mathcal{L}_{\mathrm{LAPO}} = \mathcal{L}_{\mathrm{recon}}, \quad \mathcal{L}_{\mathrm{recon}} = \|o_{t+1} - \hat{o}_{t+1}\|_2^2. \tag{2}$$

By minimizing this objective, the inferred latent action is encouraged to capture the transition information necessary to explain how the current frame evolves into the future frame, without requiring ground-truth physical actions.

b) LAOF (Bu et al., 2026): It extends the standard IDM-FDM framework by introducing optical flow as an auxiliary supervision signal for latent action learning. The motivation is that optical flow explicitly represents pixel-wise motion between consecutive frames, encouraging latent actions to focus on action-related dynamics rather than environmental distractors. Besides predicting the future frame with the FDM, LAOF also introduces a FlowDecoder to predict the target optical flow from the latent action:

$$\hat{f}_t = \mathrm{FlowDecoder}(z_t). \tag{3}$$

The overall training objective combines future-frame reconstruction with optical-flow reconstruction

$$\mathcal{L}_{\mathrm{LAOF}} = \mathcal{L}_{\mathrm{recon}} + \mathcal{L}_{\mathrm{flow}}, \quad \mathcal{L}_{\mathrm{flow}} = \|f_t - \hat{f}_t\|_2^2, \tag{4}$$

where f_t denotes the optical flow estimated from consecutive frames (o_t, o_{t+1}) by an off-the-shelf optical flow estimator.

c) CoMo (Yang et al., 2026): It is designed to mitigate shortcut learning caused by the causal leakage issue in the standard IDM-FDM framework by replacing the future frame input with the

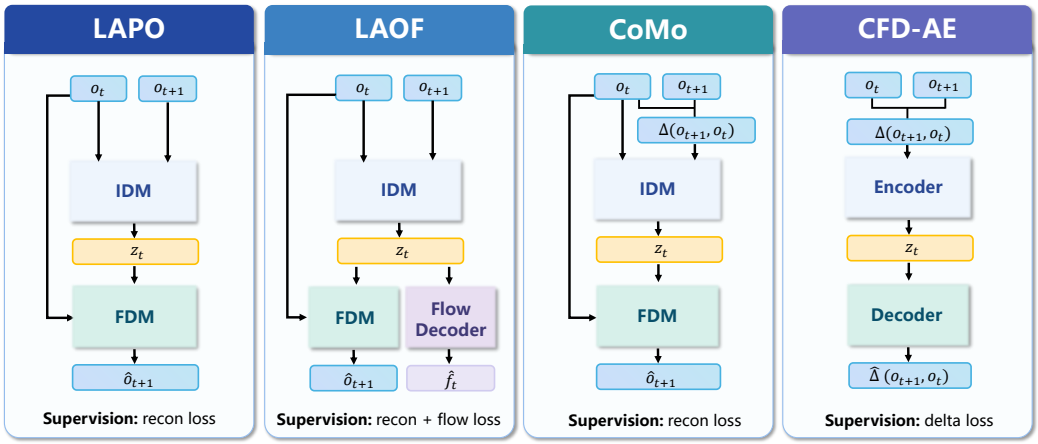

Figure 8: **Comparison of different latent action modeling paradigms.** LAPO, LAOF, and CoMo follow IDM-FDM to predict future frames, while CFD-AE reconstructs consecutive-frame differences. Arrows indicate information flow, with each panel's training losses shown at the bottom. Here, z_t denotes the latent action and Δ consecutive-frame differences; hats indicate predictions.

consecutive-frame differences. Specifically, the IDM infers the latent action from the current frame and the consecutive-frame differences Δ_t, while the FDM reconstructs the future frame from the current frame and the inferred latent action:

$$z_t = \mathrm{IDM}(o_t, \Delta_t), \quad \hat{o}_{t+1} = \mathrm{FDM}(o_t, z_t). \tag{5}$$

Following the original CoMo setting (Yang et al., 2026), we instantiate Δ_t as ΔDINO. The training objective remains the standard future-frame reconstruction loss $\mathcal{L}_{\mathrm{CoMo}} = \mathcal{L}_{\mathrm{recon}}$.

d) CFD-AE: In contrast to the IDM-FDM framework, CFD-AE directly learns latent actions by compressing and reconstructing the consecutive-frame differences Δ_t, without modeling future-frame generation:

$$z_t = \mathrm{Encoder}(\Delta_t), \quad \hat{\Delta}_t = \mathrm{Decoder}(z_t). \tag{6}$$

The training objective is simply to reconstruct the input differences representation:

$$\mathcal{L}_{\mathrm{CFD-AE}} = \mathcal{L}_{\mathrm{delta}}, \quad \mathcal{L}_{\mathrm{delta}} = \|\Delta_t - \hat{\Delta}_t\|_2^2. \tag{7}$$

B.4 Learning Objectives and Regularization

The overall optimization objective is formulated as

$$\mathcal{L} = \mathcal{L}_{\mathrm{method}} + \lambda_{\mathrm{reg}} \mathcal{L}_{\mathrm{reg}} \tag{8}$$

where $\mathcal{L}_{\mathrm{method}}$ denotes the learning objective defined by the latent action modeling method (e.g., LAPO, LAOF, CoMo, or CFD-AE), $\mathcal{L}_{\mathrm{reg}}$ denotes the latent action representation regularization term, and λ_{reg} is the regularization strength that controls the trade-off between reconstruction fidelity and latent representation regularity.

a) AE (Bank et al., 2023): It serves as the simplest baseline, where the inverse dynamics model deterministically encodes the input into a latent action z_t without any explicit latent action regularization. The model is trained solely with the method-specific objective $\mathcal{L}_{\mathrm{method}}$.

b) VQ-VAE (Van Den Oord et al., 2017): It introduces a discrete latent action regularization on the latent action inferred by the inverse dynamics model. Specifically, the latent action z_t is quantized by assigning it to its nearest codebook embedding e_k. The corresponding vector-quantization regularization is defined as

$$\mathcal{L}_{\mathrm{reg}}^{\mathrm{VQ}} = \|\mathrm{sg}[z_t] - e_k\|_2^2 + \beta_{\mathrm{vq}} \|z_t - \mathrm{sg}[e_k]\|_2^2, \tag{9}$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator and β_{vq} is the commitment weight. By constraining latent actions to a finite set of learnable codebook embeddings, VQ-VAE encourages compact, discrete, and reusable latent representations.

c) VAE (Kingma & Welling, 2013): It introduces a KL divergence regularization to constrain the conditional latent distribution $p_{\text{IDM}}(z_t \mid o_t, o_{t+1})$ toward a predefined prior distribution. The corresponding regularization objective is

$$\mathcal{L}_{\text{reg}}^{\text{VAE}} = D_{\text{KL}}\big(p_{\text{IDM}}(z_t \mid o_t, o_{t+1}) \,\|\, q(z_t)\big), \tag{10}$$

where $q(z_t) = \mathcal{N}(0, I)$ denotes a standard Gaussian prior. By minimizing the KL divergence between the inferred latent distribution and the prior, VAE encourages a smooth, continuous, and well-structured latent space.

d) Sparsity (Garrido et al., 2026): Its regularization encourages each latent action to activate only a small subset of latent dimensions, resulting in more compact and interpretable latent representations. Following (Garrido et al., 2026), we combine sample-level sparsity regularization with Variance-Covariance-Mean (VCM) regularization to avoid degenerate solutions. Given a minibatch of latent actions $Z \in \mathbb{R}^{N \times d_z}$, where each row corresponds to a latent action z_t, the regularization objective is defined as

$$\mathcal{L}_{\text{reg}}^{\text{Sparse}} = \mathcal{L}_{\text{VCM}} + \frac{1}{N}\sum_{i=1}^{N} E(Z_i), \tag{11}$$

where

$$E(z_t) = \lambda_{\ell_2} \max\Big(\sqrt{d_z} - \|z_t\|_2^2, 0\Big) + \lambda_{\ell_1}\|z_t\|_1. \tag{12}$$

The VCM regularization is given by

$$\begin{aligned}\mathcal{L}_{\text{VCM}} =& \lambda_V \frac{1}{d_z}\sum_{d=1}^{d_z} \max\Big(1 - \sqrt{\text{Var}(Z_{\cdot,d})}, 0\Big) \\ &+ \lambda_C \frac{1}{d_z(d_z-1)} \sum_{i \neq j} \text{Cov}(Z)_{i,j}^2 \\ &+ \lambda_M \frac{1}{N d_z} \sum_{i=1}^{N}\sum_{j=1}^{d_z} Z_{i,j}.\end{aligned} \tag{13}$$

This regularization encourages sparse latent action representations while preserving sufficient variance, reducing redundancy across latent dimensions, and preventing representation collapse. In practice, we follow the regularization hyperparameters in (Garrido et al., 2026) and use $\lambda_{\ell_1} = 0.01, \lambda_{\ell_2} = 1, \lambda_V = 0.1, \lambda_C = 0.001$, and $\lambda_M = 0.1$.

e) SIGReg (Balestriero & LeCun, 2025): It introduces statistical regularization of latent action representations by encouraging the latent action distribution to match an isotropic Gaussian target distribution. Directly assessing normality in high-dimensional spaces is statistically challenging, as most classical normality tests are designed for univariate distributions. Following (Balestriero & LeCun, 2025), SIGReg addresses this issue by projecting latent actions onto multiple random one-dimensional directions and matching the resulting projected distributions to a standard Gaussian. Given a minibatch of latent actions Z, SIGReg samples M random unit-norm directions $u^{(m)} \in \mathbb{S}^{d_z-1}$ and computes the projected representations $h^{(m)} = Zu^{(m)}$, $m = 1, \ldots, M$. The corresponding regularization term is defined as

$$\mathcal{L}_{\text{reg}}^{\text{SIGReg}} = \frac{1}{M}\sum_{m=1}^{M} T\Big(h^{(m)}\Big), \tag{14}$$

where $T(\cdot)$ denotes the univariate Epps–Pulley statistic (Epps & Pulley, 1983), which measures the discrepancy between each projected distribution and a standard Gaussian. Motivated by the Cramér–Wold theorem (Cramér & Wold, 1936), SIGReg encourages an isotropic Gaussian latent distribution by matching its projections along sampled random directions to a standard Gaussian. Consequently, SIGReg encourages diverse, collapse-resistant latent action representations while regularizing their distribution toward an isotropic Gaussian.

B.5 Latent Action Integration Strategies

Given the VLM backbone VLM_ψ fine-tuned with latent actions, the current observation o_t and language instruction l are first encoded into a high-level representation $h_t = \text{VLM}_\psi(o_t, l)$, which is

Table 2: Design choices explored in this work. Columns list latent action modeling paradigms, learning objectives and regularization, and latent action integration strategies. Design III distinguishes whether physical action annotations are accessible during VLM backbone fine-tuning.

Design I (●: IDM-FDM ●: CFD-AE)	**Design II** (●: Discrete ●: Continuous)	**Design III** (●: Inaccessible ●: Accessible)
● LAPO(Schmidt & Jiang, 2024)	● AE(Bank et al., 2023)	● DAP
● LAOF(Bu et al., 2026)	● VQ-VAE(Van Den Oord et al., 2017)	● LAP
● CoMo(Yang et al., 2026)	● VAE(Kingma & Welling, 2013)	● JAP
● RAFT(Teed & Deng, 2020) / SEA-RAFT(Wang et al., 2024)	● Sparsity(Garrido et al., 2026)	● JAP-DAP
● ΔRGB / ΔDINO(Oquab et al., 2024)	● SIGReg(Balestriero & LeCun, 2025)	● JAP-LAP

then passed to an action head to predict the physical action $\hat{a}_t = \mathrm{ActionHead}(h_t)$, where ψ denotes the parameters of the VLM backbone. The corresponding latent action prediction loss $\mathcal{L}_{\mathrm{latent}}$ and physical action prediction loss $\mathcal{L}_{\mathrm{act}}$ are defined as

$$\mathcal{L}_{\mathrm{latent}} = \|z_t - \hat{z}_t\|_2^2, \quad \mathcal{L}_{\mathrm{act}} = \|a_t - \hat{a}_t\|_2^2. \tag{15}$$

The three key design dimensions investigated in this work are summarized in Table 2.

a) Direct Action Prediction (DAP), $\mathrm{VLM} \rightarrow a_t$**.** During Stage II, a latent action head is attached to the VLM backbone to predict latent actions from the high-level feature representation, where $\hat{z}_t = \mathrm{ActionHead}_z(h_t)$. The VLM backbone together with the latent action head is optimized using $\mathcal{L}_{\mathrm{latent}}$, yielding a fine-tuned VLM backbone denoted by VLM_{ψ^*}. During Stage III, the latent action head is removed and replaced with a physical action head. Given $h_t = \mathrm{VLM}_{\psi^*}(o_t, l)$, the physical action is predicted as

$$\hat{a}_t = \mathrm{ActionHead}_a(h_t). \tag{16}$$

The model is then optimized using $\mathcal{L}_{\mathrm{act}}$ on robot action data.

b) Latent-to-Action Prediction (LAP), $\mathrm{VLM} \rightarrow z_t \rightarrow a_t$**.** Stage II is the same as in DAP. During Stage III, the fine-tuned VLM backbone and the latent action head are inherited from Stage II. Given $h_t = \mathrm{VLM}_{\psi^*}(o_t, l)$, instead of directly predicting the physical action from h_t, the model first predicts a latent action and subsequently decodes it into the physical action:

$$\hat{z}_t = \mathrm{ActionHead}_z(h_t), \quad \hat{a}_t = \mathrm{ActionHead}_a(\hat{z}_t). \tag{17}$$

The VLM backbone together with the latent action head and physical action head is jointly optimized using $\mathcal{L}_{\mathrm{act}}$ on robot action data.

c) Joint Latent-Action Prediction (JAP), $\mathrm{VLM} \rightarrow [z_t, a_t]$**.** A shared action head simultaneously predicts latent and physical actions:

$$[\hat{z}_t, \hat{a}_t] = \mathrm{ActionHead}_{z,a}(h_t). \tag{18}$$

Unlike DAP and LAP, which first fine-tune the VLM backbone using latent actions before downstream policy learning, JAP assumes access to robot action data and jointly optimizes latent and physical action prediction in a single training stage. Therefore, no separate latent-action-only VLM fine-tuning stage is required. During Stage III, the VLM backbone together with the joint action head is optimized using the objective $\mathcal{L} = \mathcal{L}_{\mathrm{latent}} + \mathcal{L}_{\mathrm{act}}$, where $\mathcal{L}_{\mathrm{latent}}$ is computed on video data and $\mathcal{L}_{\mathrm{act}}$ is computed on robot action data.

d) JAP-DAP and JAP-LAP. The VLM backbone is first jointly fine-tuned with both latent actions and physical actions following the JAP setting. The trained action head is then discarded, and a new DAP or LAP head is trained from scratch. Both the joint fine-tuning and subsequent training with the new action head take place in Stage III.

C Experiments

This appendix provides experimental details and supplementary discussion for Section 3.

C.1 Experimental Setup

Datasets. The corpus described in Section 3.1 consists of a subset of the open-source Open X-Embodiment (OXE)(O'Neill et al., 2024) real-world robot datasets, including DROID (Khazatsky et al., 2024) (45.6%), Language Table (Lynch et al., 2023) (11.7%), BC-Z (Jang et al., 2021) (9.4%), Furniture Bench (Heo et al., 2023) (6.7%), Fractal (Brohan et al., 2022) (6.7%), Bridge (Walke et al., 2023) (3.3%), FMB (Luo et al., 2024) (1.7%), and Stanford Hydra (Belkhale et al., 2023) (0.6%), together with the simulated datasets Robotwin (10.6%) and liberoplus (3.9%). The liberoplus videos used in Stages I and II cover all perturbation types, including RobotInit. For the RoboTwin2.0 benchmark, we use the complete Robotwin dataset, which contains 50 bimanual manipulation tasks collected on the Aloha-AgileX robot platform, with each task including 50 clean demonstrations and 500 randomized demonstrations. For the LIBERO and LIBERO-Plus benchmarks, Stage-III policy training uses only the Libero datasets, including Libero-10, Libero-goal, Libero-object, and Libero-spatial, and the policies are evaluated on LIBERO-Plus under RobotInit without further Stage-III training. Notably, to ensure a fair comparison using the same training data, we compute optical flow directly from the raw data. In particular, we retain optical flow estimates for the Noise setting in LIBERO-Plus without applying any setting-specific filtering.

LIBERO(Liu et al., 2023). It is a single-arm manipulation benchmark designed to evaluate compositional generalization and long-horizon task execution over shared objects and environments. We evaluate on 4 task suites, namely {Goal, Object, Spatial, and Long}, which respectively test generalization over task goals, manipulated objects, spatial relations, and temporally extended instruction following.

LIBERO-Plus(Fei et al., 2025). It extends LIBERO by introducing diverse robustness perturbations to evaluate whether learned policies overfit to specific environmental conditions. It includes seven perturbation types: {Camera, RobotInit, Noise, Language, Layout, Light, Background}. The evaluation protocol in Section 3.1 evaluates policy robustness to variations in robot initialization while preserving the original task objectives and evaluation setup.

RoboTwin2.0(Chen et al., 2025a). It is a dual-arm manipulation benchmark designed to evaluate cross-domain generalization in bimanual settings. The benchmark provides paired environments with similar task structures but different visual appearances and dynamics. Due to computational constraints, we evaluate on 12 representative tasks with a horizon of 400 steps, selected from the full set of 50 tasks. These tasks include 6 easier tasks and 6 more challenging tasks for controlled ablations. Specifically, the selected tasks are {Adjust Bottle, Lift Pot, Move Playingcard Away, Move Stapler Pad, Pick Diverse Bottles, Place Container Plate, Place Mouse Pad, Place Object Stand, Place Phone Stand, Press Stapler, Rotate QRCode, and Turn Switch}.

Implementation Details. The IDM is composed of 24 encoder blocks for latent action inference, while the FDM is composed of 24 decoder blocks for future-frame prediction. To capture consecutive-frame motion representations across multiple temporal scales, input videos are temporally downsampled using a random factor sampled from $\{1, 2, 3, 4, 5\}$. All video frames are center-cropped and resized to a fixed resolution of 224×224. We optimize the model using AdamW (Loshchilov & Hutter, 2017), with detailed training hyperparameters provided in Table 3.

C.2 Which latent action modeling paradigms produce the highest-quality latent action representations? (**Q1**)

We provide further discussion of the comparisons in Section 3.2. Although ΔRGB and optical flow both capture changes between consecutive frames, ΔRGB directly encodes the pixel-wise difference between the two frames, whereas optical flow further transforms these changes into pixel-wise displacements, imposing a stronger motion-centric inductive bias. Such a representation may discard visual cues relevant to transition dynamics, including contact, occlusion and disocclusion, local deformation, and object boundary changes, which cannot always be adequately characterized by pixel-wise displacements. Moreover, optical flow introduces an additional motion estimation process, through which estimation errors may propagate to the learned latent action representation and reduce its robustness to visual perturbations. Visual perturbations such as noise may also affect optical flow estimation. Although SEA-RAFT achieves better average performance than

Table 3: Training hyperparameters for the three training stages. Stage I trains LAM, Stage II finetunes the VLM backbone with latent actions, and Stage III trains the downstream policy. Superscripts identify benchmark-specific settings.

Hyperparameter	Stage I	Stage II	Stage III
Base LR	2.5×10^{-5}	2.5×10^{-5}	2.5×10^{-5}
Action Head LR	–	1×10^{-4}	1×10^{-4}
Qwen3-VL-4B LR	–	1×10^{-5}	1×10^{-5}
LR Scheduler	Constant	Cosine with Min LR	Cosine with Min LR
Weight Decay	1×10^{-2}	1×10^{-8}	1×10^{-8}
Global Batch Size	256	256	128^{+} / 32^{-} / 32^{*}
Training Steps	150k	150k	80k^{+} / 150k^{-} / 40k^{*}
Action Chunk Size	-	1	8^{+} / 50^{-} / 50^{*}

LR: Learning Rate; $^{+}$: LIBERO / LIBERO-Plus; $^{-}$: RoboTwin2.0; *: Real-World.

RAFT on LIBERO, RAFT outperforms SEA-RAFT on LIBERO-Plus under distribution shifts. This comparison suggests that the choice of optical flow estimator can affect downstream performance differently across benchmarks.

Although ΔRGB naturally suppresses static backgrounds by differencing consecutive frames in pixel-wise space, it contains little semantic information about the underlying visual state transitions. In contrast, ΔDINO computes frame differences in a pretrained semantic feature space, where low-level visual details are partially abstracted into higher-level representations of objects and scenes, making it more suitable for robotic manipulation.

C.3 How do regularization methods and their regularization strengths affect latent action learning? (**Q2**)

Fig. 2. We select the regularization strength λ_{reg} using a composite metric defined as $\text{SSIM Gain} + \text{MSE Gain} - \text{MLP} - \text{Linear}$, with the chosen value indicated by the red vertical lines. Excessively strong regularization suppresses transition dynamics, leading to degraded future-frame reconstruction and weaker physical action prediction. This trend is particularly evident in the FDM reconstruction metrics, where overly large regularization strengths consistently reduce both SSIM Gain and MSE Gain for VAE, Sparsity, and SIGReg. Meanwhile, the probing metrics, $-\text{Linear probe}$ and $-\text{MLP probe}$, also deteriorate under inappropriate regularization strengths.

For the LIBERO-Plus result in Section 3.3, a possible explanation is that the discrete bottleneck encourages the model to capture reusable high-level action primitives rather than overfit to fine-grained continuous variations in the source domain. Such action primitives may transfer more effectively under environmental perturbations, which is particularly beneficial for zero-shot evaluation. This advantage does not generalize across benchmarks. VQ-VAE achieves comparable performance on LIBERO but lower average performance on RoboTwin2.0 than the other evaluated methods. While discretization may improve robustness to distribution shifts, it also reduces the continuity and flexibility of latent actions, making it more difficult to further improve performance on benchmarks where performance is already saturated. Sparsity introduces additional hyperparameters and SIGReg incurs additional computational overhead from sampling random projection directions. These considerations support the default choice of VAE in Section 3.3.

C.4 How should latent actions be integrated into physical action prediction? (**Q3**)

We supplement the overall comparison in Section 3.4 with the following results from Table 1.

Physical Action Inaccessible. The two exceptions to LAP outperforming DAP are LAOF (0.6563 vs. 0.6873) and VQ-VAE (0.7477 vs. 0.7560), with each value averaged over the three bench-

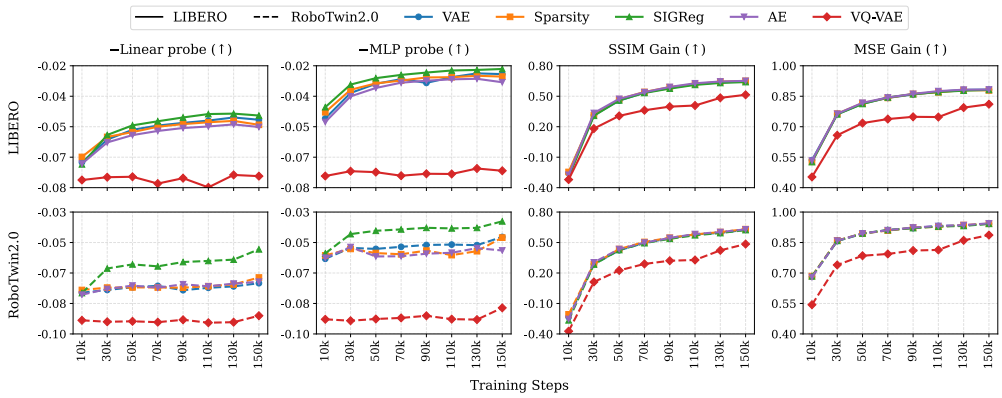

Figure 9: **Performance of different regularization methods throughout training.** The four columns plot the proxy metrics defined in Section 3.5 against training steps, up to 150K. The top and bottom rows show LIBERO (solid lines) and RoboTwin2.0 (dashed lines), respectively. Colors distinguish AE and the four regularization methods. Higher values indicate better proxy scores.

marks. These exceptions show that retaining latent actions as an intermediate representation does not improve physical action prediction for every latent action configuration.

The comparison with JAP in Section 3.4 concerns overall mean performance across the evaluated configurations. The similar overall means do not imply that LAP and JAP perform similarly for each configuration. Averaging over the three benchmarks, LAP performs better than JAP with ΔDINO and VAE, while JAP performs better with RAFT and LAOF. This distinction is relevant when choosing a strategy for a specific latent action model.

Physical Action Accessible. Both JAP-DAP and JAP-LAP achieve higher mean success rates than JAP on each of the three benchmarks when averaged over the 12 configurations. Their relative performance differs between benchmarks. Among these three strategies, JAP-LAP achieves the highest mean success rate on LIBERO (0.8964) and LIBERO-Plus (0.4243), while JAP-DAP achieves the highest mean on RoboTwin2.0 (0.8450), compared with 0.8428 for JAP-LAP. These benchmark-specific comparisons supplement the overall ranking in Section 3.4 and explain why the preferred variant can depend on the target benchmark.

C.5 Are proxy metrics for latent actions predictive of downstream robotic manipulation performance? (**Q4**)

Fig. 4. The *Regularization methods* setting in Fig. 4 comprises 21 experimental configurations, including one AE configuration; four regularization strengths, $\lambda_{\text{reg}} \in \{10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}\}$, evaluated for each of VAE, Sparsity, SIGReg, and VQ-VAE; four additional configurations corresponding to the regularization strengths selected in this work, namely VAE (10^{-7}), Sparsity (10^{-5}), SIGReg (10^{-3}), and VQ-VAE (1). The + *Latent action dimensionality* setting further augments these configurations with eight configurations with different latent action dimensionalities, $d_z \in \{8, 16, 32, 64, 128, 256, 512, 1024\}$, for 29 experimental configurations in total. For each configuration, we report results under DAP and LAP, as these two settings train the VLM backbone solely using latent actions without physical action supervision in Stage II, thereby providing a direct measure of the correlation between proxy metrics of latent actions and downstream robotic manipulation performance.

These correlations summarize agreement across the evaluated configurations and do not directly measure how accurately a proxy metric selects the best-performing model. In particular, a positive correlation does not guarantee that the highest proxy score corresponds to the highest downstream success rate. This distinction supports the screening recommendation in Section 3.5, with final model selection based on downstream task evaluation.

As dimensionality increases, latent actions can encode progressively richer information, which may reduce probe prediction errors even when such gains in representational capacity do not translate

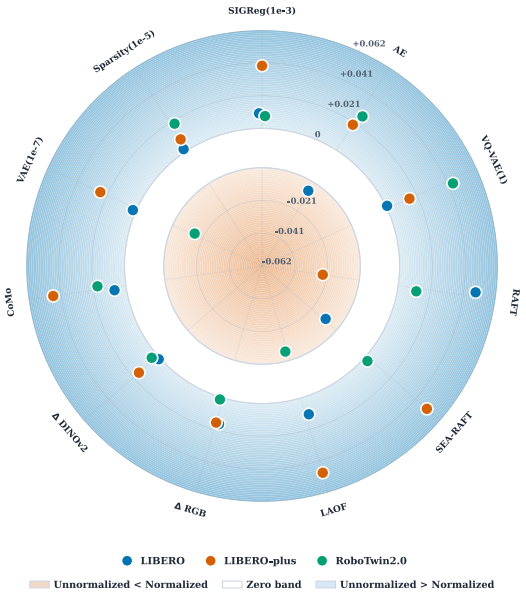

Figure 10: **Performance difference between unnormalized and normalized settings across methods and benchmarks.** Angular groups identify methods, with regularization strengths in parentheses; radial position shows the success-rate gap (Unnormalized − Normalized), averaged over DAP and LAP. Blue, orange, and green points denote LIBERO, LIBERO-Plus, and RoboTwin2.0, respectively. The white annulus denotes the zero band; the outer blue and inner orange regions indicate gains and losses without normalization, respectively.

into improved downstream performance. At the same time, higher dimensional representations may facilitate shortcut learning for frame reconstruction, thereby weakening the discriminative power of reconstruction metrics.

The lower probe losses on LIBERO discussed in Section 3.5 and shown in Fig. 9 may partly reflect differences in action parameterization. However, LIBERO and RoboTwin2.0 also differ in action dimensionality and task distribution, and the comparability of their probe losses depends on action scaling and loss normalization. Latent actions derived from consecutive-frame motion may be more aligned with differential control formulations. In contrast, position control requires predicting or recovering absolute target states, which may be more task-dependent and may contribute to higher probe losses. The four proxy metrics generally improve over training. This trend suggests that the learned latent action representations progressively encode more action-relevant information, even under joint position control, potentially indicating the emergence of a more structured motion representation space during training.

C.6 What is the most suitable latent action dimensionality, and is latent action normalization necessary? (**Q5**)

Fig. 5. Since LIBERO is relatively prone to performance saturation and potential overfitting, the marginal differences reported in Section 3.6 are unlikely to reflect meaningful differences in generalization. The LIBERO-Plus results suggest that lower-dimensional latent spaces may help suppress redundant variations for relatively simple single-arm manipulation tasks. The RoboTwin2.0 results suggest that excessively small latent dimensions are insufficient to capture the complexity of bimanual manipulation, whereas excessively large latent spaces do not consistently improve performance and may instead introduce redundant or less structured representations.

For the normalization comparison in Section 3.6 and Fig. 10, a plausible explanation is that regularization has already imposed sufficient structure and stability on the latent space, which may limit the benefit of additional normalization; normalization may also suppress magnitude-related information relevant to downstream policy learning.

C.7 Does fine-tuning the VLM backbone with latent actions exhibit scaling laws in robotic manipulation? (**Q6**)

Fig. 6. With the Stage III training data, algorithm, model architecture, and optimization hyperparameters kept identical, this comparison examines how changing the Stage-II fine-tuning corpus affects the VLM backbone initialization for downstream policy learning. Specifically, the training data, algorithm, model architecture, and all optimization hyperparameters in Stage III are kept identical, while the Stage-II VLM backbone is fine-tuned using either `Robotwin+liberoplus` (14.5% of the full dataset) or `Robotwin+liberoplus+OXE` (100%). The improvements for both DAP and LAP reported in Section 3.7 show that the benefit is observed with both direct physical action prediction from VLM features and prediction through latent actions. Thus, the observed benefit is not restricted to retaining latent actions as an intermediate representation during Stage III.

C.8 Real-World Experiments

Fig. 7. Specifically, we design four representative tabletop manipulation tasks: `Close Drawer`, involving closing the top red drawer; `Open Drawer`, involving opening the top red drawer; `Stack Bowls`, requiring two bowls to be stacked; and `Place Block`, requiring a block to be placed into a bowl. We compare OpenVLA-OFT, which is trained using the original Qwen3-VL-4B backbone, and OpenVLA-OFT (LA-Tuned), which uses a Qwen3-VL-4B backbone fine-tuned with latent actions on 59M video frames annotated with latent actions using LAPO with VAE (10^{-7}) regularization, a latent action dimensionality of $d_z = 32$, and unnormalized latent actions. For each task, 50 robot demonstrations are collected under joint position control. To ensure a fair comparison, both models follow the same Stage III training configuration (Table 3) and are each trained on the same combined dataset of 200 robot demonstrations collected across the four manipulation tasks. Furthermore, OpenVLA-OFT (LA-Tuned) uses DAP as the latent action integration strategy, ensuring that the only difference between two models is the parameters of the VLM backbone. We evaluate checkpoints at 6k, 8k, 10k, 20k, and 40k training steps. Each checkpoint is evaluated over 20 independent trials, resulting in 100 evaluations per task for each method. All experiments use only a single front-view RGB camera as the visual input.

During evaluation, each policy is deployed on the real robot in a closed-loop manner, with task success rate serving as the primary evaluation metric. The rollouts start from randomized initial states, resetting the scene before every rollout. To further increase evaluation diversity, we randomize both the initial object configurations and the placement of task-irrelevant distractor objects on the tabletop, while ensuring that the same set of evaluation scenes is used for all methods. A rollout is considered successful only if the robot completes the target manipulation within a predefined execution horizon.

Specifically, take the `Open Drawer` task as an example, the robot arm must grasp the round handle with a relatively large UMI gripper. This requires the gripper to sufficiently wrap around the handle while avoiding collisions with the upper cabinet surface. OpenVLA-OFT (LA-Tuned) exhibits better performance in this task. During evaluation, we further observe that OpenVLA-OFT (LA-Tuned) exhibits stronger robustness to task-irrelevant distractors. For example, when objects from other tasks are placed in the scene, such as bowls appearing during the drawer-opening task, the LA-Tuned model can still focus on the task-relevant object and complete the intended manipulation. In contrast, the baseline OpenVLA-OFT is more likely to switch to an incorrect task mode and interact with the distractor object. This suggests that latent-action tuning not only improves manipulation success rates, but also enhances the task-conditioned grounding ability of the VLM backbone, enabling the policy to better distinguish task-relevant affordances from irrelevant visual distractors.