FRESHVLA: FREQUENCY-REGULARIZED TIME-SCALE HIERARCHICAL LATENT ACTIONS FOR VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning reusable latent actions from action-free videos offers an effective route to improving Vision-Language-Action (VLA) policies. However, reconstruction-oriented latent actions do not necessarily benefit policy learning: as they may encode task-irrelevant static appearance with scene identity, or entangle slowly varying task progress with fine-grained execution dynamics in limited discrete code. We propose **FreshVLA**, a VLA framework built on a frequency-guided, dynamics-sensitive learning paradigm for acquiring temporally hierarchical continuous latent actions. FreshVLA uses a high-pass prior to guide visual transition encoding toward dynamical regions, thereby reducing reliance on static appearance cues. It further organizes latent actions into coarse-grained task progress and fine-grained execution dynamics through cross-timescale sampling and frequency-domain regularization, enhancing their capacity to represent long-horizon behaviors. During policy training, we freeze the pretrained latent action encoder and transfer its representations to robot control through joint flow-matching of latent and robot actions. FreshVLA achieves 98.2% average success on LIBERO and 98.0% on LIBERO-Long, exceeding GR00T-N1.6 on the latter by 3.6%. It also reaches an average sequence length of 4.5 on CALVIN and strong overall success rate of 80.4% in the zero-shot LIBERO-PLUS evaluation. Furthermore, representation analysis reveals more consistent robot–object interaction attention in FreshVLA and a 51.3% greater relative error reduction in long-horizon action regression than discrete latent actions. Real-world experiments demonstrate that FreshVLA improves the average success rate over $\pi_{0.5}$ by 7.5%.

Keywords: latent actions, vision-language-action models, temporal frequency

1 INTRODUCTION

Vision-Language-Action (VLA) models aim to translate language-conditioned visual understanding into executable robot behavior. Recent systems have advanced this paradigm through larger multimodal backbones, broader robot datasets, and more expressive action decoders (Zitkovich et al., 2023; Ghosh et al., 2024; Kim et al., 2025b; Black et al., 2025). Nevertheless, a fundamental interface challenge remains: policies must transform visual intent into temporally coherent control behavior. This requires representing the visual transitions that should occur in a scene, including which object should move, how contact relationships evolve, and how local execution supports longer-horizon task progress.

Latent Action Models (LAMs) provide a natural means of learning such representations. By predicting or reconstructing observation transitions, LAMs infer latent actions from adjacent frames or observation sequences (Ye et al., 2025; Liang et al., 2026a; Bu et al., 2025). These variables can provide supervision for policy pretraining while reducing the need to directly align raw action spaces across embodiments. However, reconstruction objectives do not guarantee that latent actions capture controllable and task-relevant changes. They may instead encode background layouts, static appearance, object identity, or other cues that help predict future observations but are only weakly related to control (Bu et al., 2026b; Cai et al., 2025; Bu et al., 2026a). Furthermore, manipulation trajectories contain both slowly evolving task progress, such as approaching and transporting an ob-

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.

ject, and local execution dynamics, such as grasp adjustments and release timing. Many prior LAMs represent fixed-interval visual transitions with vector-quantized codes (van den Oord et al., 2017; Ye et al., 2025; Bu et al., 2025), or continuous formulations with single latent transition scale (Liang et al., 2026a; Yang et al., 2026), which may obscure this multi-timescale structure and hinder downstream policies from modeling the temporal dependencies of long-horizon manipulation.

To address aforementioned challenges, we propose **FreshVLA** to construct dynamic-sensitive hierarchical latent actions from visual transitions, as is described in Figure 1. In the visual feature space, temporal high-pass responses provide localized cues of change and guide the patch-level attention module toward regions exhibiting task-relevant visual dynamics. In the latent action space, temporal downsampling and upsampling, together with temporal-frequency regularization, encourage the substage-scale transition to model persistent long-horizon dependencies, while the execution-scale representation captures immediate dynamic residuals. Both representations are further used in bidirectional future prediction to explain changes in visual observations. During VLA learning, the pretrained latent action model is kept frozen to preserve the transition space learned from action-free videos. Conditioned on visual-language context, a latent-action flow-matching mechanism jointly models latent actions and robot actions. (Lipman et al., 2023; Li et al., 2026).

With 2.5B parameters, FreshVLA achieves an average success rate of 98.2% on LIBERO, reaches 98.0% on LIBERO-Long, and attains an average sequence length of 4.5 on CALVIN. These results exceed those of GR00T-N1.6 (NVIDIA GEAR Team, 2025) by 1.2 and 3.6 percentage points on the two LIBERO metrics, respectively, and by 0.26 subtasks on CALVIN, highlighting its strength in long-horizon manipulation. In zero-shot LIBERO-PLUS evaluation, it achieves 80.4% macro-average success and outperforms OpenVLA-OFT (Kim et al., 2025a) across all seven perturbation categories, reaching 88.9%, 97.0%, and 97.1% under language, lighting, and background shifts, respectively. Representation analyses show consistent attention to end-effector–object interaction regions, while an action-regression probe finds a 51.3% greater relative error reduction than discrete latent actions. Across four real-world tasks, FreshVLA improves mean success by 7.5 percentage points over $\pi_{0.5}$, suggesting more reliable object interaction and long-horizon execution.

Our contributions are summarized as follows:

- We introduce a temporally hierarchical continuous latent action representation that organizes visual transitions into substage-scale components capturing long-horizon task progress and execution-scale residuals preserving local dynamics.

- We propose **FreshVLA**, which learns this representation using a high-pass visual prior and frequency-domain regularization, then transfers it to VLA policy learning through joint latent-action flow matching with a frozen latent action encoder.

- FreshVLA performs strongly across LIBERO, CALVIN, and zero-shot LIBERO-PLUS, achieving 98.0% on LIBERO-Long and an average sequence length of 4.5 on CALVIN, highlighting its long-horizon performance. In real-world tasks, it exceeds $\pi_{0.5}$ by 7.5% in average success rate.

2 Related Work

2.1 Vision–Language–Action Models

Vision–Language–Action (VLA) models map multimodal context—including visual observations, language instructions, and available proprioceptive states—to executable robot control. RT-1 and RT-2 established scalable language-conditioned robot-policy learning (Brohan et al., 2023; Zitkovich et al., 2023). Subsequent generalist policies scale this paradigm by integrating heterogeneous robot data or pretrained multimodal backbones with specialized action generators (O'Neill et al., 2024; Ghosh et al., 2024; Kim et al., 2025b; Black et al., 2025; NVIDIA et al., 2025). Despite their improved generality, these policies are trained to predict robot actions whose coordinates and control conventions remain embodiment dependent (O'Neill et al., 2024; Bu et al., 2025). At the level of labeled action representation, FAST applies DCT-based compression for efficient tokenization of high-frequency action chunks (Pertsch et al., 2025), whereas MINT decomposes action chunks in frequency space into coarse intent and fine execution (Huang et al., 2026). While these methods improve the modeling of labeled robot actions, they remain embodiment-specific and re-

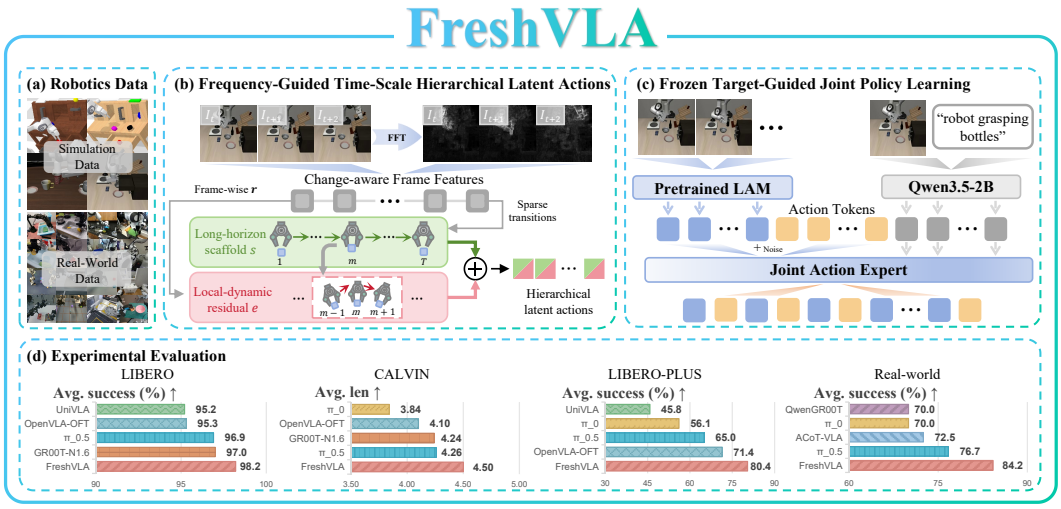

Figure 1: FreshVLA uses a high-pass prior to extract dynamics-sensitive features and organizes them into hierarchical latent actions that capture long-horizon task progress. For policy learning, a frozen latent action encoder guides the joint denoising of latent and robot actions. Evaluations in simulation, representation analyses, and real-world manipulation demonstrate its effectiveness.

quire action supervision. This motivates learning embodiment-agnostic, control-oriented representations from action-free videos.

2.2 Latent Action Learning

Action-free videos provide object-motion and task-progress cues at scale without robot action annotations (Ye et al., 2025; Yang et al., 2026). Latent Action Models (LAMs) convert such visual changes into action-like variables, typically by inferring compact transition representations from observation pairs and using them for prediction or downstream control. Discrete approaches commonly employ VQ-VAE-style quantization (van den Oord et al., 2017; Bruce et al., 2024). LAPA applies such codes to VLA pretraining (Ye et al., 2025), while UniVLA derives task-centric latent actions for cross-embodiment policy learning (Bu et al., 2025). Other work explores continuous latent transitions, representation alignment, and transfer from heterogeneous human videos (Liang et al., 2026a; Yang et al., 2026; Liu et al., 2026; Dai et al., 2026; Zhang et al., 2026; Luo et al., 2026). Related formulations introduce image-goal, motion-token, distilled, unified world-model, or 3D-aware representations (Chen et al., 2024; 2025; Li et al., 2025; Bi et al., 2026; Govind et al., 2026; Liang et al., 2026b), while a 2026 study evaluates 41 latent action design choices spanning modeling, objectives, and downstream integration (Bu et al., 2026a). FreshVLA leverages a high-pass prior to guide transition encoding toward dynamically changing regions, while cross-timescale construction and DCT-based regularization organize latent actions into substage-scale task progress and execution-scale local dynamics before embodiment-specific policy alignment.

3 Method

3.1 Overview and Problem Setup

A basic latent action model consists of an inverse dynamics model $\mathcal{E}$ and a forward dynamics model $\mathcal{D}$ (Ye et al., 2025; Liang et al., 2026a). Given consecutive observations o_t and o_{t+1}, the inverse dynamics model extracts a latent transition $\tilde{z}_{t\to t+1}$, while the forward dynamics model predicts the next observation conditioned on the current observation and the inferred transition:

$$\tilde{z}_{t\to t+1} = \mathcal{E}(o_t, o_{t+1}), \qquad \hat{o}_{t+1} = \mathcal{D}(o_t, \tilde{z}_{t\to t+1}), \qquad \mathcal{L}_{\text{LAM}} = \|\hat{o}_{t+1} - o_{t+1}\|_2^2 . \tag{1}$$

Because pronounced ego-motion from wrist-mounted cameras can be easily confounded with genuine interaction dynamics, FreshVLA currently pretrains its latent actions on fixed third-person videos, providing a stable observation coordinate frame.

3.2 STAGE I: FREQUENCY-GUIDED HIERARCHICAL LATENT TARGETS

Dynamics-Sensitive Visual Transition Encoding. As discussed above, LAMs may place excessive attention on static information such as background appearance, resulting in predictive shortcuts and misleading action representations. We therefore introduce a high-pass response prior to guide the model toward dynamically changing regions involved in manipulation. As illustrated in Figure 2, Given an observation window $\mathbf{o}_{1:T}$, a frozen DINOv3 visual encoder (Siméoni et al., 2025) F extracts patch-level tokens $\mathbf{U}_t$. A trainable DynaAdapter A_ψ aggregates these tokens into frame-level representations $\mathbf{h}_t$ and outputs normalized patch-attention distributions $\mathbf{w}_t$. We then compute frame-level transition features by differencing adjacent frame-level representations:

$$\mathbf{U}_t = F(\mathbf{o}_t), \quad (\mathbf{h}_t, \mathbf{w}_t) = A_\psi(\mathbf{U}_t), \quad \mathbf{d}_t = \mathbf{h}_{t+1} - \mathbf{h}_t. \tag{2}$$

Specifically, we apply a real-valued FFT (Cooley & Tukey, 1965) along the temporal axis at each RGB pixel and retain frequency bins $k \geq \lfloor 0.5(\lfloor T/2 \rfloor + 1) \rfloor$. The inverse FFT yields frame-wise high-pass responses, whose magnitudes are normalized and aggregated across RGB channels by taking the maximum. After spatial pooling onto the patch grid, the resulting scores are normalized to form target distributions $\boldsymbol{\pi}_t$, with gradients stopped through the target branch. For a batch of B observation windows indexed by b, the Patch-Attention Weight KL Alignment loss is defined as

$$\mathcal{L}_{\mathrm{PAWKL}} = \frac{1}{BT} \sum_{b=1}^{B} \sum_{t=1}^{T} D_{\mathrm{KL}} \left(\mathrm{sg}(\boldsymbol{\pi}_{b,t}) \,\|\, \mathbf{w}_{b,t} \right). \tag{3}$$

PAWKL encourages the learned attention distribution to approach the temporal high-pass target, thereby reducing the visual aggregator's reliance on static appearance and directing attention toward regions undergoing visual change.

Hierarchical Latent Action Construction. Many existing latent action models rely on flat VQ codebooks (Ye et al., 2025; Bu et al., 2025), which limit their ability to express long-horizon relationships, so we introduce a temporally hierarchical continuous representation. Let $N = T - 1$ denote the number of frame-level transitions in an observation window of T frames. Approximately equally spaced temporal anchors $\alpha_1 < \cdots < \alpha_{K+1}$ divide each observation window into K coarse intervals. The substage-scale input summarizes the accumulated feature change within each interval:

$$\mathbf{d}_k^{\mathrm{sub}} = \mathbf{h}_{\alpha_{k+1}} - \mathbf{h}_{\alpha_k}, \qquad k = 1, \ldots, K. \tag{4}$$

We use a shared temporal encoder to project both transition types into a common hidden space and model temporal dependencies at different scales. The substage transition encoder R_s captures long-horizon task dependencies and is then upsampled to construct a residual signal. The execution transition encoder R_e processes this residual to obtain the execution-scale representation:

$$\begin{aligned} \mathbf{r} &= E_c(\mathbf{d}_{1:N}), \qquad \mathbf{s} = \mathrm{Up}_N \left(R_s \left(E_c(\mathbf{d}_{1:K}^{\mathrm{sub}}) \right) \right), \\ \mathbf{e} &= R_e(\mathbf{r} - \mathbf{s}), \qquad \mathbf{z} = \mathrm{LN}_{\mathrm{out}}(\mathbf{s} + \mathbf{e}). \end{aligned} \tag{5}$$

Overall, Eq. 5 defines the latent action encoder $\mathcal{H}$: $\mathbf{z} = \mathcal{H}(\mathbf{d}_{1:N}, \mathbf{d}_{1:K}^{\mathrm{sub}})$. Here, $\mathbf{r}, \mathbf{s}, \mathbf{e}, \mathbf{z} \in \mathbb{R}^{B \times N \times D}$, where D is the feature dimension. Specifically, $\mathbf{r}$ denotes the encoded frame-level transition relationship, $\mathbf{s}$ the upsampled substage-scale representation, $\mathbf{e}$ the execution-scale representation, and $\mathbf{z}$ the composite hierarchical latent action.

To explicitly encourage the temporal-scale structure in Eq. 5, we align $\mathbf{s}$ and $\mathbf{z}$ with $\mathbf{r}$ in the DCT domain (Ahmed et al., 1974). The substage alignment uses the first K temporal frequency bins, encouraging $\mathbf{s}$ to capture slowly varying components of $\mathbf{r}$. We align $\mathbf{z}$ over all N bins to promote fidelity across temporal frequencies, with scale-specific frequency weights defined below. For two such sequences $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{B \times N \times D}$, let $\mathcal{C}(\cdot)$ denote the orthogonal DCT applied along the temporal axis independently for each feature channel. We define the weighted discrepancy as

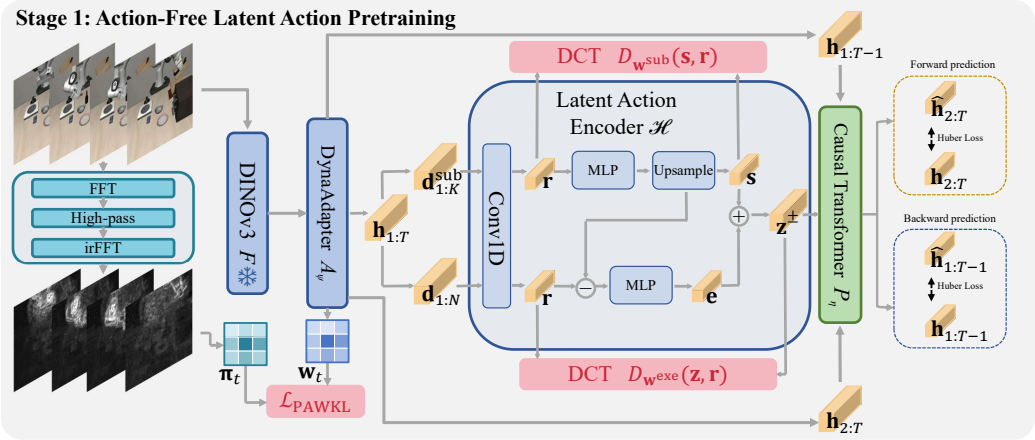

Figure 2: Illustration of the pretraining architecture. DINOv3 and DynaAdapter first extract dynamics-sensitive visual transitions under FFT-based PAWKL guidance. The latent action encoder $\mathcal{H}$ then constructs temporally hierarchical continuous latent actions, while DCT regularization encourages scale-specific temporal structure, with Causal Transformer as a bidirectional predictor.

$$D_{\boldsymbol{\omega}}(\mathbf{x},\mathbf{y}) = \frac{1}{BD}\sum_{b=1}^{B}\sum_{\ell=1}^{D}\sum_{j=0}^{N-1}\omega_j\left|\mathcal{C}(\mathbf{x})_{bj\ell} - \mathcal{C}(\mathbf{y})_{bj\ell}\right|^2. \tag{6}$$

The temporal DCT maps each length-N transition sequence to N frequency bins indexed by $j = 0,\ldots,N-1$. The substage weights $\boldsymbol{\omega}^{\text{sub}}$ assign normalized exponentially decaying weights to the first K bins, whereas the execution weights $\boldsymbol{\omega}^{\text{exe}}$ apply the same decay rule over all N bins. To learn transitions in both temporal directions, we apply the shared hierarchical encoder in Eq. 5 to the original and sign-reversed transition features. The resulting forward and reverse representations are denoted by $(\mathbf{r}^\sigma,\mathbf{s}^\sigma,\mathbf{e}^\sigma,\mathbf{z}^\sigma)$, where $\sigma\in\{+,-\}$. The hierarchical encoding objective is

$$\mathcal{L}_{\mathcal{H}} = \frac{1}{2}\sum_{\sigma\in\{+,-\}}\left[\lambda_{\text{sub}}D_{\boldsymbol{\omega}^{\text{sub}}}(\mathbf{s}^\sigma,\mathbf{r}^\sigma) + \lambda_{\text{exe}}D_{\boldsymbol{\omega}^{\text{exe}}}(\mathbf{z}^\sigma,\mathbf{r}^\sigma)\right]. \tag{7}$$

Bidirectional Prediction Objective. The forward and reverse latent actions are given by

$$\mathbf{z}^+ = \mathcal{H}\left(\mathbf{d},\mathbf{d}^{\text{sub}}\right), \qquad \mathbf{z}^- = \mathcal{H}\left(-\mathbf{d},-\mathbf{d}^{\text{sub}}\right). \tag{8}$$

A shared causal Transformer P_η predicts adjacent frame representations conditioned on hierarchical latent actions. For reverse prediction, $\text{rev}(\cdot)$ temporally reverses inputs, latents, and targets. The training losses are

$$\mathcal{L}_{\text{dyn}} = \frac{1}{2}\Big[\ell_{\text{Huber}}\left(P_\eta(\mathbf{h}_{1:T-1},\mathbf{z}^+),\mathbf{h}_{2:T}\right) + \ell_{\text{Huber}}\left(P_\eta(\text{rev}(\mathbf{h}_{2:T}),\text{rev}(\mathbf{z}^-)),\text{rev}(\mathbf{h}_{1:T-1})\right)\Big],$$
$$\mathcal{L}_{\text{total}} = \lambda_{\text{d}}\mathcal{L}_{\text{dyn}} + \lambda_{\text{h}}\mathcal{L}_{\mathcal{H}} + \lambda_{\text{p}}\mathcal{L}_{\text{PAWKL}} + \lambda_{\text{s}}\mathcal{L}_{\text{SIG}}. \tag{9}$$

The prediction objective uses the Huber loss (Huber, 1964). Here, $\mathcal{L}_{\text{SIG}}$ is the Sketched Isotropic Gaussian regularizer introduced in LeJEPA (Balestriero & LeCun, 2025), applied to the frame-level representations $\mathbf{h}_{1:T}$. It encourages the empirical characteristic function of randomly projected features to approach that of a standard Gaussian distribution, thereby mitigating representation collapse.

3.3 Stage II: Joint Latent-Action Flow Matching

Building on flow-matching (Lipman et al., 2023), we apply the hierarchical latent actions to VLA policy through a joint latent action and robot action training paradigm. After Stage I, we freeze the

visual encoder F, DynaAdapter A_ψ, and latent action encoder $\mathcal{H}$ as a target generator G. Given an observation window, G produces forward hierarchical latent actions:

$$G(\mathbf{o}_{t:t+T-1}) = \mathcal{H}\left(\mathbf{d}_{1:N}, \mathbf{d}^{\text{sub}}_{1:K}\right) = \mathbf{z}^{+}. \tag{10}$$

Let $H_{\text{a}} \leq N$ denote the action horizon. The first H_{a} latent actions, $\mathbf{z}^{+}_{1:H_{\text{a}}}$, are aligned with the corresponding robot-action chunk $\mathbf{a}_{t:t+H_{\text{a}}-1}$. The policy is conditioned on the current observation, language instruction, and robot state, denoted by $\mathbf{c}_t = (\mathbf{o}_t, l, \mathbf{q}_t)$. Both streams are independently perturbed with Gaussian noise and jointly processed by a shared flow-matching action expert:

$$\widetilde{\mathbf{z}}_{\tau,i} = (1-\tau)\boldsymbol{\epsilon}^z_i + \tau \mathbf{z}^{+}_i, \qquad \widetilde{\mathbf{a}}_{\tau,i} = (1-\tau)\boldsymbol{\epsilon}^a_i + \tau \mathbf{a}_{t+i-1}, \qquad i = 1, \ldots, H_{\text{a}}. \tag{11}$$

Here, $\boldsymbol{\epsilon}^z_i$ and $\boldsymbol{\epsilon}^a_i$ refer to standard Gaussian noise. The action expert predicts velocity fields $\widehat{\mathbf{v}}^a_{\tau,i}$ and $\widehat{\mathbf{v}}^z_{\tau,i}$ for the robot-action and latent-action streams, respectively, with the objective

$$\mathcal{L}_{\text{JFM}} = \mathbb{E}\left[\frac{1}{H_{\text{a}}}\sum_{i=1}^{H_{\text{a}}}\left(\text{MSE}\left(\widehat{\mathbf{v}}^a_{\tau,i}, \mathbf{a}_{t+i-1} - \boldsymbol{\epsilon}^a_i\right) + \text{MSE}\left(\widehat{\mathbf{v}}^z_{\tau,i}, \mathbf{z}^{+}_i - \boldsymbol{\epsilon}^z_i\right)\right)\right]. \tag{12}$$

Joint denoising transfers visual-dynamics information from the latent action stream to continuous robot action prediction. During inference, G and future observations are unnecessary. Both streams are initialized with Gaussian noise and jointly denoised, while only the resulting robot action chunk is executed; the latent stream remains an internal variable.

4 Experiments

4.1 Experimental Setup

Dataset and Training Configuration. We construct a large-scale pretraining corpus of 227,894 trajectories from diverse real and simulated environments. For latent action pretraining, hierarchical continuous latent actions are extracted from 36-frame observation windows. For downstream policy training, we use Qwen3.5-2B (Qwen Team, 2026) as the vision-language backbone and a DiT-based joint flow-matching action expert to generate 7-D robot actions. See Appendix A.2 for architecture details and Appendices A.1 and A.3 for data and training configurations.

4.2 Simulation Experiments

LIBERO. We evaluate FreshVLA on LIBERO (Liu et al., 2023), a widely used language-conditioned robotic manipulation suite for assessing policy generalization under diverse task variations. Following the standard protocol, we report results on four suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. We compare against LAPA (Ye et al., 2025), OpenVLA (Kim et al., 2025b), OpenVLA-OFT (Kim et al., 2025a), VILLA-X (Chen et al., 2026), UniVLA (Bu et al., 2025), π_0 (Black et al., 2025), $\pi_{0.5}$ (Physical Intelligence et al., 2025), and GR00T-N1.6 (NVIDIA GEAR Team, 2025). As shown in Table 1, FreshVLA achieves an average success rate of 98.2% with 2.5B parameters and outperforms these strong baselines overall. Although FreshVLA underperforms on LIBERO-Goal, likely because its latent action pretraining does not explicitly condition on language goals, it remains highly competitive overall. Notably, FreshVLA achieves a 98.0% success rate on LIBERO-Long, improving over UniVLA and GR00T-N1.6 by 6.0 and 3.6, respectively. This result suggests that temporally hierarchical latent actions better preserve the compositional structure required for long-horizon manipulation.

LIBERO-PLUS. To further evaluate generalization under distribution shifts, we assess FreshVLA on LIBERO-PLUS (Fei et al., 2026). Built upon the standard LIBERO suites, LIBERO-PLUS is a robustness benchmark that introduces perturbations along seven dimensions. As shown in Table 2, FreshVLA achieves the best performance across all seven perturbation dimensions, with a macro-average success rate of 80.4%, outperforming other baselines. In particular, FreshVLA shows clear advantages under Language, Light, Background, and Noise perturbations. These results suggest

Table 1: Performance comparison on LIBERO and CALVIN. LIBERO reports success rates (%). CALVIN reports success rates (%) after completing 1–5 consecutive subtasks and the average achieved sequence length.

Method	Size	LIBERO					CALVIN					
		Spa.	Obj.	Goal	Long	Avg.	1	2	3	4	5	Len.
LAPA (Ye et al., 2025)	7B	73.8	74.6	58.8	55.4	65.7	–	–	–	–	–	–
OpenVLA (Kim et al., 2025b)	7B	84.7	88.4	79.2	53.7	76.5	91.3	77.8	62.0	52.1	43.5	3.27
OpenVLA-OFT (Kim et al., 2025a)	7B	96.2	98.3	96.2	90.7	95.3	96.3	89.1	82.4	75.8	66.5	4.10
VILLA-X (Chen et al., 2026)	3B	97.5	97.0	91.5	74.5	90.1	–	–	–	–	–	–
UniVLA (Bu et al., 2025)	9B	96.5	96.8	95.6	92.0	95.2	95.5	85.8	75.4	66.9	56.5	3.80
π_0 (Black et al., 2025)	3B	96.8	98.8	95.8	85.2	94.2	93.8	85.0	76.7	68.6	60.1	3.84
$\pi_{0.5}$ (Physical Intelligence, 2025)	3B	**98.8**	98.2	**98.0**	92.4	96.9	94.4	88.4	85.3	80.1	76.1	4.26
GR00T-N1.6 (NVIDIA GEAR Team, 2025)	3B	97.7	98.5	97.5	94.4	97.0	96.3	91.4	85.3	79.7	71.7	4.24
FreshVLA (Ours)	2.5B	98.7	**99.7**	96.3	**98.0**	**98.2**	**99.5**	**96.3**	**91.2**	**85.1**	**77.9**	**4.50**

Table 2: LIBERO-PLUS robustness results (%) under zero-shot manner.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Avg.
OpenVLA (Kim et al., 2025b)	0.8	3.5	23.0	8.1	34.8	15.2	28.5	16.3
OpenVLA-OFT (Kim et al., 2025a)	56.4	31.9	79.5	88.7	93.3	75.8	74.2	71.4
UniVLA (Bu et al., 2025)	1.8	46.2	69.6	69.0	81.0	21.2	31.9	45.8
π_0 (Black et al., 2025)	13.8	6.0	58.8	85.0	81.4	79.0	68.9	56.1
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	53.0	50.3	65.7	83.1	77.3	53.2	72.7	65.0
FreshVLA (Ours)	60.0	58.9	88.9	97.0	97.1	84.4	76.3	**80.4**

that the robustness of the learned latent actions to visual and semantic variations transfers to the downstream policy, thereby improving its out-of-distribution generalization.

CALVIN. To evaluate a model's ability to execute multiple robot manipulation tasks consecutively, we evaluate FreshVLA on CALVIN (Mees et al., 2022). Following the standard protocol, each instruction sequence consists of five consecutive subtasks. As shown in Table 1, FreshVLA achieves the best success rate at every sequence length, reaching 99.5%, 96.3%, 91.2%, 85.1%, and 77.9% after completing 1–5 subtasks, while its first-subtask success rate exceeds $\pi_{0.5}$, by 5.1%. FreshVLA also achieves the highest average completion length of 4.5. These results suggest that its temporally hierarchical latent actions help sustain reliable execution across extended task sequences.

4.3 Representation Analysis Experiments

Qualitative Experiments. We visualize the visual attention patterns of V-JEPA 2.1 (Mur-Labadia et al., 2026), LAPA (Ye et al., 2025), UniVLA (Bu et al., 2025), and FreshVLA by extracting model-specific attention weights over spatial visual tokens and mapping them onto frame-wise heatmaps. This allows us to compare how the models attend to task-relevant objects, interaction regions. As shown in Figure 3, V-JEPA 2.1 attends broadly to the scene, including background regions and objects unrelated to the interaction. LAPA and UniVLA focus more on end-effector–object interactions, but their attention shifts noticeably across consecutive frames. In contrast, FreshVLA more consistently tracks the interacting object as it moves. This evidence suggests a more temporally coherent, interaction-aware representation that may better support long-horizon manipulation.

Quantitative Evaluation of Long-Horizon Gains. To further analyze how latent action structure affects long-horizon task expression, we conduct an action-regression probe to measure influence of different context on regression error. We freeze the latent action encoders of all four variants: SCLA/SDLA (single-layer continuous/discrete) and HCLA/HDLA (hierarchical continuous/discrete), then independently train an action regressor (Yang et al., 2026; Alain & Bengio, 2016) on its latent representations. All regressors use the same transformer architecture, parameter count, data split, and optimization protocol, thereby controlling for differences in probe capacity. A local bidirectional mask limits the visible context to H_c steps. On held-out trajectories from LIBERO benchmark, let $E(H_c)$ denote the held-out action-prediction MSE when the probe observes H_c la-

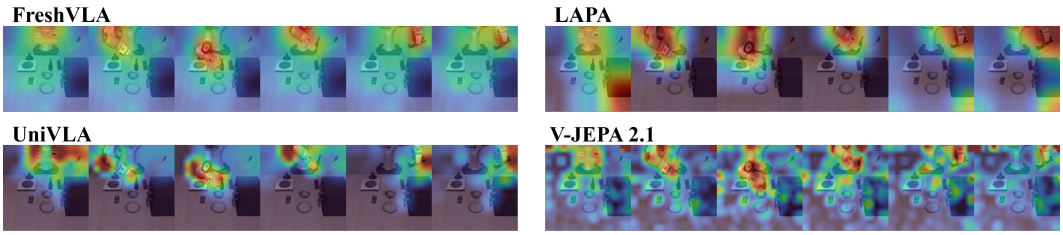

Figure 3: Spatial attention maps across consecutive frames for FreshVLA, LAPA, UniVLA, and V-JEPA 2.1, highlighting differences in spatial focus and temporal consistency during manipulation.

Suite	SDLA	HDLA	SCLA	**HCLA (Ours)**
Long	92.0	72.3	93.0	**98.0**
Goal	96.0	96.0	95.7	**96.3**
Object	96.7	93.3	99.7	**99.7**
Spatial	95.3	90.7	95.7	**98.7**
Avg.	95.0	88.1	96.0	**98.2**

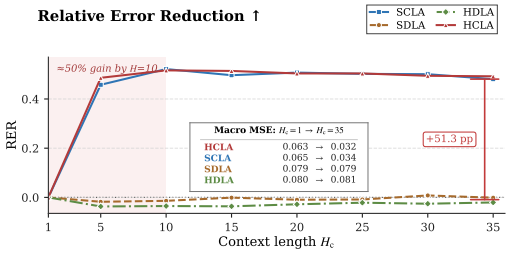

Figure 4: **Left:** Ablation study of latent action structures on LIBERO. **Right:** Action-regression probe showing Relative Error Reduction (RER) versus context length H_c.

tent steps. Taking the single-step error $E^{\text{base}} := E(1)$ as the baseline, we define

$$\text{RER}(H_c) = \frac{E^{\text{base}} - E(H_c)}{E^{\text{base}}}. \tag{13}$$

As shown in Figure 4(right), SDLA and HDLA improve only marginally as H_c increases, whereas SCLA and HCLA exhibit lower action-prediction error even at $H_c = 1$ and benefit markedly from additional context. At $H_c = 35$, HDLA incurs a substantially higher MSE than HCLA and achieves an RER that is 51.3% lower, suggesting that continuous latent actions support more effective context-dependent action decoding. Despite similar RER trends, HCLA achieves approximately 5.9% lower MSE than SCLA at $H_c = 35$, indicating an additional benefit from temporal hierarchy.

4.4 Real-World Experiments

To evaluate FreshVLA's transferability to real-world environments, we conduct experiments on a 6-DoF Flexiv robotic arm equipped with RealSense D435 cameras, as shown in Figure 7. We evaluate four language-conditioned tabletop manipulation tasks spanning basic to long-horizon operations. For each task, we collect 30 trajectories and train four baselines under the same setting. The QwenGR00T baseline draws on Qwen3.5 (Qwen Team, 2026) and a GR00T-style action expert (NVIDIA et al., 2025); ACoT-VLA (Zhong et al., 2026) and $\pi_{0.5}$ (Physical Intelligence et al., 2025) are selected as strong reasoning- and long-horizon-oriented baselines, respectively. As reported in Table 3, FreshVLA achieves an average success rate of 84.2%, outperforming ACoT-VLA and $\pi_{0.5}$ by 11.7 and 7.5 percentage points, respectively. On Warmup Dinner, which requires precise object interaction, FreshVLA presses the bell's raised button more reliably and achieves a higher success rate. We hypothesize that its latent actions' focus on interaction-relevant visual dynamics contributes to this improvement. FreshVLA also achieves success rates of 86.7% and 73.3% on the multistage, multiobject Organize Bowls and Collect Toys tasks, respectively. These results support the utility of temporally hierarchical continuous latent actions for multistage, contact-rich manipulation in real-world settings.

4.5 Ablation Study

Latent Action Structure Analysis. We compare four latent action configurations: SCLA/SDLA (single-layer continuous/discrete) and HCLA/HDLA (hierarchical continuous/discrete). The hierarchical variants use the temporal downsampling, upsampling, and residual decomposition described in equation 5, whereas the discrete variants use VQ codebook to compress latent action. As shown

Table 3: Real-world manipulation success rates (%) across four tasks.

Method	Open Pot	Warmup Dinner	Organize Bowls	Collect Toys	Avg.
π_0	96.7	63.3	53.3	66.7	70.0
QwenGR00T	93.3	60.0	66.7	60.0	70.0
ACoT-VLA	96.7	70.0	56.7	66.7	72.5
$\pi_{0.5}$	96.7	66.7	80.0	63.3	76.7
FreshVLA (Ours)	**96.7**	**80.0**	**86.7**	**73.3**	**84.2**

Table 4: Ablation of frequency-decay weighting, latent actions, and PAWKL on LIBERO. We report success rates (%) for each task suite and their average.

Ablation Factor	Variant	Long	Goal	Object	Spatial	Avg.
FR Decay Weight	0 (Uniform)	96.0	**98.3**	99.7	98.3	98.1
	1 (Exponential)	**98.0**	96.3	99.7	**98.7**	**98.2**
	2	96.0	96.3	**100.0**	98.3	97.7
Latent Action	w/o	93.0	96.3	96.0	98.7	96.0
PAWKL	w/o	95.3	98.0	99.3	98.3	97.7

in Figure 4 (left), HCLA achieves the best average success rate of 98.2%. Within the continuous setting, temporal hierarchy improves LIBERO-Long performance by 5.0% over the single-scale counterpart, supporting its utility for long-horizon manipulation.

Frequency-Regularization Decay. We vary the decay used by frequency regularization to study its effect on downstream success. The decay parameter controls the relative loss weights assigned to low- and high-frequency components through the slope of an exponential decay. As shown in the Table 4, the standard exponential decay setting achieves the best average success rate of 98.2% and the highest LIBERO-Long performance of 98.0%. Uniform weighting performs comparably, while excessive decay weakens goal-oriented performance, indicating that suppressing high-frequency execution details too strongly may impair fine-grained manipulation.

Effectiveness of Latent Actions. As shown in Table 4, removing latent actions reduces the average success rate from 98.2% to 96.0% on LIBERO, with a 5.0-point drop on LIBERO-Long (98.0% $\rightarrow$ 93.0%). These results support the contribution of latent actions to long-horizon manipulation.

Effectiveness of PAWKL Divergence. To assess PAWKL's contribution to downstream performance, we compare FreshVLA variants trained with and without this regularizer on LIBERO. As shown in Table 4, PAWKL increases the average success rate from 97.7% to 98.2%, with gains of 2.7%, 0.4%, and 0.4% on LIBERO-Long, LIBERO-Object, and LIBERO-Spatial, respectively. Performance on LIBERO-Goal decreases slightly, possibly because greater emphasis on interaction regions reduces sensitivity to goal-location cues. Overall, the results are consistent with PAWKL directing attention toward manipulation-relevant changes, particularly benefiting long-horizon tasks.

5 Conclusion

We present **FreshVLA**, a frequency-guided, dynamics-sensitive framework for learning temporally hierarchical continuous latent actions for VLA policies. FreshVLA uses a high-pass prior to emphasize manipulation-relevant visual changes, while cross-timescale construction and frequency regularization encourage complementary representations of task progress and execution dynamics. For downstream control, a joint flow-matching policy denoises latent action and robot action streams using targets from a frozen latent action encoder. With 2.5B parameters, FreshVLA achieves 98.2% average success on LIBERO, 98.0% on LIBERO-Long, and an average sequence length of 4.5 on CALVIN. On zero-shot LIBERO-PLUS, it attains 80.4% overall success and outperforms OpenVLA-OFT across all seven perturbation categories. Representation analyses show more consistent attention to end-effector–object interactions and lower long-horizon action-regression error. In real-world tasks, FreshVLA exceeds $\pi_{0.5}$ by 7.5 percentage points in average success rate.

AI USE STATEMENT

We used generative AI tools to assist with format conversion, English translation of selected passages that were present only in the Chinese draft, and LaTeX compilation troubleshooting. The authors reviewed the converted manuscript, checked the retained experimental values and equations against the draft, and take responsibility for the final content.

REPRODUCIBILITY STATEMENT

We describe the two-stage training framework and learning objectives in Section 3 and the experimental setup in Section 4. Additional implementation details are provided in Appendix A, including the pretraining dataset composition and provenance (Appendix A.1), latent action model architecture (Appendix A.2), and training pipelines and optimization settings (Appendix A.3). Appendix B.1 details the evaluation setups and metrics for the additional representation analyses.

REFERENCES

Nasir Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974. doi: 10.1109/T-C.1974.223784. URL https://doi.org/10.1109/T-C.1974.223784.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025. URL https://arxiv.org/abs/2511.08544.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35101–35113, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Bi_Motus_A_Unified_Latent_Action_World_Model_CVPR_2026_paper.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert 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. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.010. URL https://doi.org/10.15607/RSS.2025.XXI.010.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S Ryoo, Grecia Salazar, Pannag R Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan H Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics Transformer for Real-World Control at Scale. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.025.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623, 2024. URL https://arxiv.org/abs/2402.15391.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Learning to Act Anywhere with Task-centric Latent Actions. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.014.

Xizhou Bu, Qingda Hu, Lei Zhou, Lingfeng Zhang, Yingbo Tang, Zihao Liu, Xinyi Tao, Zhiqiang Ma, Qingqiu Huang, Chufeng Tang, Hongbo Wang, Jing Zhang, Jiayi Ma, Hangjun Ye, Wei Li, and Xiaoshuai Hao. What matters for latent actions in robot learning. *arXiv preprint arXiv:2608.19613*, 2026a. URL https://arxiv.org/abs/2608.19613.

Xizhou Bu, Jiexi Lyu, Fulei Sun, Ruichen Yang, Zhiqiang Ma, and Wei Li. LAOF: Robust latent action learning with optical flow constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27334–27344, June 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/html/Bu_LAOF_Robust_Latent_Action_Learning_with_Optical_Flow_Constraints_CVPR_2026_paper.html.

Zhejia Cai, Yandan Yang, Xinyuan Chang, Shiyi Liang, Ronghan Chen, Feng Xiong, Mu Xu, and Ruqi Huang. Seeing space and motion: Enhancing latent actions with geometric and dynamic awareness for vision-language-action models. *arXiv preprint arXiv:2509.26251*, 2025. URL https://arxiv.org/abs/2509.26251.

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. URL https://arxiv.org/abs/2411.00785.

Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, Jianyu Chen, and Jiang Bian. VILLA-X: Enhancing latent action modeling in vision-language-action models. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2507.23682.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19752–19763, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Chen_Moto_Latent_Motion_Token_as_the_Bridging_Language_for_Learning_ICCV_2025_paper.html.

James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. doi: 10.1090/S0025-5718-1965-0178586-1. URL https://doi.org/10.1090/S0025-5718-1965-0178586-1.

Weisheng Dai, Kai Lan, Jianyi Zhou, Bo Zhao, Xiu Su, Junwen Tong, Weili Guan, and Shuo Yang. ConLA: Contrastive latent action learning from human videos for robotic manipulation. *arXiv preprint arXiv:2602.00557*, 2026. URL https://arxiv.org/abs/2602.00557.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38574–38583, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Fei_LIBERO-Plus_A_Progressive_Robustness_Benchmark_for_Visual-Language-Action_Models_CVPR_2026_paper.html.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.090. URL https://doi.org/10.15607/RSS.2024.XX.090.

Manish Kumar Govind, Dominick Reilly, Pu Wang, and Srijan Das. UniLACT: Depth-aware RGB latent action learning for vision-language-action models. *arXiv preprint arXiv:2602.20231*, 2026. URL https://arxiv.org/abs/2602.20231.

Stijn Hawinkel, Willem Waegeman, and Steven Maere. Out-of-sample R^2: Estimation and inference. *The American Statistician*, 78(1):15–25, 2024. doi: 10.1080/00031305.2023.2216252. URL https://doi.org/10.1080/00031305.2023.2216252.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Chengkai Hou, Kun Wu, Jiaming Liu, Zhengping Che, Di Wu, Fei Liao, Guangrun Li, Jingyang He, Qiuxuan Feng, Zhao Jin, Chenyang Gu, Zhuoyang Liu, Nuowei Han, Xiangju Mi, Yaoxu Lv, Yankai Fu, Gaole Dai, Langzhe Gu, Tao Li, Yuheng Zhang, Yixue Zhang, Xinhua Wang, Shichao Fan, Meng Li, Zhen Zhao, Ning Liu, Zhiyuan Xu, Pei Ren, Junjie Ji, Haonan Liu, Kuan Cheng, Shanghang Zhang, and Jian Tang. RoboMIND 2.0: A multimodal, bimanual mobile manipulation dataset for generalizable embodied intelligence, 2026. URL https://arxiv.org/abs/2512.24653.

Renming Huang, Chendong Zeng, Wenjing Tang, Jintian Cai, Cewu Lu, and Panpan Cai. Mimic intent, not just trajectories. *arXiv preprint arXiv:2602.08602*, 2026. URL https://arxiv.org/abs/2602.08602.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. doi: 10.1214/aoms/1177703732. URL https://doi.org/10.1214/aoms/1177703732.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.120. URL https://doi.org/10.15607/RSS.2024.XX.120.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025a. doi: 10.15607/RSS.2025.XXI.017. URL https://doi.org/10.15607/RSS.2025.XXI.017.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025b. URL https://proceedings.mlr.press/v270/kim25c.html.

Qiwei Li, Xicheng Gong, Xinghang Li, Peiyan Li, Quanyun Zhou, Hangjun Ye, Jiahuan Zhou, and Yadong Mu. RotVLA: Rotational latent action for vision-language-action model. *arXiv preprint arXiv:2605.13403*, 2026. URL https://arxiv.org/abs/2605.13403.

Zuolei Li, Xingyu Gao, Xiaofan Wang, and Jianlong Fu. LatBot: Distilling universal latent actions for vision-language-action models. *arXiv preprint arXiv:2511.23034*, 2025. URL https://arxiv.org/abs/2511.23034.

Anthony Liang, Pavel Czempin, Matthew M. Hong, Yutai Zhou, Jingzhen Wang, Erdem Biyik, and Stephen Tu. CLAM: Continuous latent action models for robot learning from unlabeled demonstrations. In *2026 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026a. URL https://arxiv.org/abs/2505.04999.

Qiwei Liang, Boyang Cai, Minghao Lai, Sitong Zhuang, Tao Lin, Yan Qin, Yixuan Ye, Jiaming Liang, and Renjing Xu. Bootstrap dynamic-aware 3D visual representation for scalable robot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13419–13429, June 2026b.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Mengya Liu, Baoxiong Jia, Jiangyong Huang, Jingze Zhang, and Siyuan Huang. LARA: Latent action representation alignment for vision-language-action models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Hao Luo, Ye Wang, Wanpeng Zhang, Haoqi Yuan, Yicheng Feng, Haiweng Xu, Sipeng Zheng, and Zongqing Lu. Joint-aligned latent action: Towards scalable VLA pretraining in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35047–35058, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Luo_Joint-Aligned_Latent_Action_Towards_Scalable_VLA_Pretraining_in_the_Wild_CVPR_2026_paper.html.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022. doi: 10.1109/LRA.2022.3180108.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026. URL https://arxiv.org/abs/2603.14482.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. URL https://arxiv.org/abs/2503.14734.

NVIDIA GEAR Team. GR00T N1.6: An improved open foundation model for generalist humanoid robots, December 2025. URL https://research.nvidia.com/labs/gear/gr00t-n1_6/. Official technical report.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, et al. Open X-embodiment: Robotic learning datasets and RT-X models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

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. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.012.

Physical Intelligence. OpenPI: LIBERO benchmark evaluation. Official OpenPI evaluation documentation, 2025. URL https://github.com/Physical-Intelligence/openpi/blob/main/examples/libero/README.md. Accessed 2026-08-23.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5. Official model report.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. URL https://arxiv.org/abs/2508.10104.

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

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1711.00937.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on robot learning*, pp. 1723–1736. PMLR, 2023.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 42352–42363, June 2026.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.11758.

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. URL https://arxiv.org/abs/2601.04061.

Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Si Liu, and Guanghui Ren. ACoT-VLA: Action chain-of-thought for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8152–8162, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhong_ACoT-VLA_Action_Chain-of-Thought_for_Vision-Language-Action_Models_CVPR_2026_paper.html.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

SUPPLEMENTARY MATERIAL

A IMPLEMENTATION DETAILS

A.1 DATASET COMPOSITION DETAILS

To pretrain temporally hierarchical latent actions, we construct a diverse corpus using the LeRobot framework. The resulting mixture comprises 227,894 episodes and roughly 27.87 million frames. The detailed statistics of our pretraining data mixture are provided in Table 5.

For real-world interaction dynamics, we aggregate three prominent datasets from the Open X-Embodiment (OXE) (O'Neill et al., 2024) repository. Specifically, DROID (Khazatsky et al., 2024) provides the largest share of visual dynamics with 74,740 episodes and 19.82 million frames. RT-1 (Brohan et al., 2023) contributes an additional 86,397 episodes with 3.72 million frames, while Bridge (Walke et al., 2023) supplies 28,935 episodes containing 982 thousand frames. To further broaden the coverage of real-world manipulation, we incorporate data from RoboMIND2 (Hou et al., 2026) and RoboCOIN (Wu et al., 2025). We sample episodes across all their constituent sub-datasets. This approach yields 2,621 episodes with 203 thousand frames for RoboMIND2, and 15,638 episodes with 1.79 million frames for RoboCOIN.

For structured and temporally dense dynamics, we include two simulated benchmarks: CALVIN (Mees et al., 2022) with 17,870 tabletop episodes and 1.07 million frames, and LIBERO (Liu et al., 2023) with 1,693 language-conditioned episodes and 273 thousand frames. Ultimately, this mixture exposes the model to heterogeneous temporal scales: from short, dense simulated transitions to long-tailed real-world trajectories, which is essential for learning both fine-grained execution and long-horizon compositional structures.

Table 5: Detailed overview of the pretraining dataset mixture. The corpus aggregates real-world and simulated manipulation data to provide heterogeneous temporal scales for latent action learning.

Metric	DROID	RT-1	Bridge	RoboCOIN	RoboMIND2	CALVIN	LIBERO	Total
Episodes	74,740	86,397	28,935	15,638	2,621	17,870	1,693	227,894
Frames	19,823,040	3,724,262	982,895	1,790,845	203,049	1,071,743	273,465	27,869,299

A.1.1 ROBOCOIN TASK-LEVEL BREAKDOWN AND PROVENANCE

To ensure rigorous data provenance and full reproducibility, we detail the 65 distinct task-level repositories sampled from the RoboCOIN dataset in Table 8. Because RoboCOIN aggregates datasets from various contributors, licenses are maintained at the individual repository level on Hugging Face. All utilized repositories are fully licensed under Apache-2.0.

A.1.2 ROBOMIND 2.0 SUBSET-LEVEL BREAKDOWN AND PROVENANCE

In addition to RoboCOIN, our pretraining mixture incorporates 2,621 trajectories from the RoboMIND 2.0 dataset. We utilize 16 official release subsets downloaded from the ModelScope platform. During our data processing pipeline, the five disjoint parts of the Franka subset are unified into a single training group, resulting in 12 distinct training groups. To ensure strict data provenance, Table 9 details the specific subsets and their corresponding trajectory counts. All 16 subsets are licensed under Apache-2.0 as declared in their respective ModelScope repository metadata.

A.2 LATENT ACTION ARCHITECTURE DETAILS

Network Architecture. The core architectural hyperparameters for our latent action model are summarized in Table 6. Specifically, the DynaAdapter A_ψ aggregates spatial patches into frame-level representations $\mathbf{h}_t$ via cross-attention, which is supervised by the PAWKL loss using high-pass filtered frequency components. The hierarchical continuous encoder $\mathcal{H}$ utilizes symmetric padding in its 1D temporal convolution to integrate bidirectional temporal context, while the substage-scale representation is aligned with the low-frequency DCT bins via $\mathcal{L}_\mathcal{H}$.

Table 6: Detailed architectural hyperparameters for the latent action model.

Module	Hyperparameter	Value
Inputs & Backbone	Observation window size & Resolution	36 frames, 224×224
	Visual backbone	DINOv3 ViT-B/16 (Frozen)
DynaAdapter A_ψ	Output token dimension	768
	Cross-attention mechanism	4 heads, single shared query
Hierarchical Encoder $\mathcal{H}$	Latent-action space dimension	64
	Substage count (K)	6
	Temporal convolution (Conv1D)	Kernel size 3 (symmetric padding)
Bidirectional Predictor P_η	Architecture	Causal Transformer
	Layers & Hidden dimension	3 layers, 1024

Sketched Isotropic Gaussian Regularization (SIGReg). Let $\mathbf{h} \in \mathbb{R}^{B\times T\times D}$ denote the frame-level representations. At each time step, SIGReg computes empirical characteristic functions across the batch for randomly sampled, unit-norm projection directions. For direction $\mathbf{a}_m$, define

$$\begin{aligned} \widehat{\phi}_{t,m}(\tau_q) &= \frac{1}{B}\sum_{b=1}^{B} \exp\big(i\tau_q\, \mathbf{a}_m^\top \mathbf{h}_{b,t}\big)\,, \\ \mathcal{L}_{\mathrm{SIG}} &= \frac{1}{TM}\sum_{t=1}^{T}\sum_{m=1}^{M} B \sum_{q=0}^{Q-1} \omega_q \left|\widehat{\phi}_{t,m}(\tau_q) - \exp\left(-\frac{\tau_q^2}{2}\right)\right|^2 . \end{aligned} \tag{14}$$

Here, $\mathbf{a}_m \in \mathbb{S}^{D-1}$ is a sampled unit-norm direction, i is the imaginary unit, and

$$\begin{aligned} &\tau_q = \frac{3q}{Q-1}, \qquad \Delta\tau = \frac{3}{Q-1}, \\ &\omega_q = c_q\, \Delta\tau\, \exp\left(-\frac{\tau_q^2}{2}\right), \qquad c_q = \begin{cases} 1, & q = 0 \text{ or } q = Q-1, \\ 2, & \text{otherwise.} \end{cases} \end{aligned} \tag{15}$$

The weights implement trapezoidal quadrature over $\tau_q \in [0, 3]$, using the even symmetry of the Epps–Pulley integrand. The implementation uses $Q = 17$ quadrature points and $M = 1024$ projection directions.

A.3 Training Pipelines and Optimization

Implementation Framework. Our downstream policy is built upon the StarVLA framework (StarVLA Community, 2026), utilizing Qwen3.5-2B (Qwen Team, 2026) as the visual-language backbone to extract multimodal context. All models are trained and evaluated on a single computation node equipped with 8 NVIDIA RTX 4090 GPUs.

Stage I: Latent Action Pretraining. The upstream model processes 36-frame windows with a substage resolution of $K = 6$. We optimize the network for 120,000 steps using AdamW (Loshchilov & Hutter, 2019) with a total batch size of 128 and a base learning rate of 10^{-4}.

Stage II: Joint Flow-Matching Policy. During downstream policy training, the pretrained latent action encoder remains frozen. The DiT-B action expert receives an 8-dimensional robot proprioceptive state and an interleaved sequence of continuous latent action and 7-dimensional robot-action tokens, $\big(z_i^{\mathrm{token}}, a_i^{\mathrm{token}}\big)_{i=1}^{12}$, with an action horizon of $H_{\mathrm{a}} = 12$. This stage is optimized with AdamW at a base learning rate of 10^{-4}.

B Additional Results

B.1 Additional Representation Analysis

B.1.1 Quantitative Evaluation on Visual Robustness.

We study whether latent actions retain action-decodable information under visual shifts. Following LIBERO-PLUS, we perturb only the third-person-view images through background, illumination, or

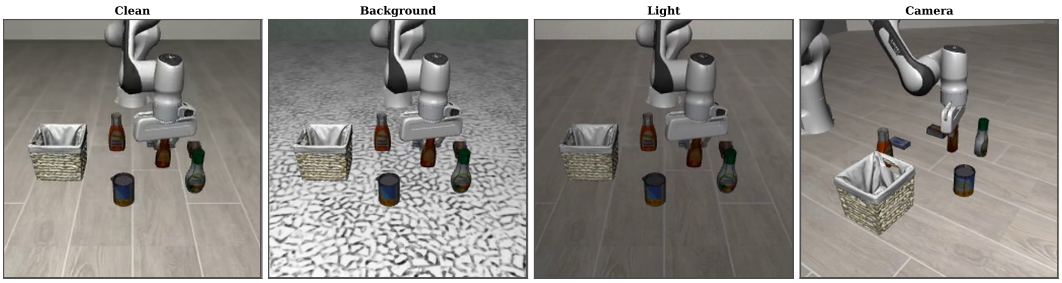

Figure 5: Examples of visual perturbations used in the robustness evaluation, including variations in background, illumination, and camera viewpoint.

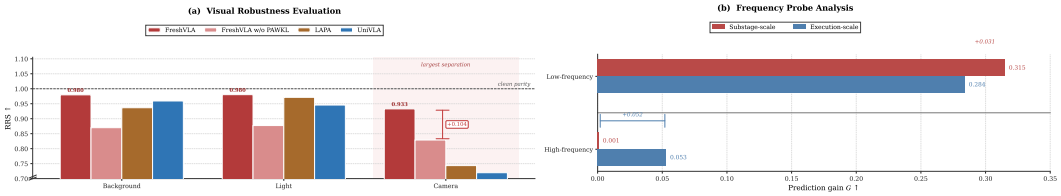

Figure 6: Visual robustness and frequency probes. (a) Robustness of FreshVLA variants and latent action baselines to background, illumination, and camera perturbations. (b) Macro predictive R^2 for low/high DCT coefficients decoded from substage- and execution-scale representations; brackets show point-estimate differences.

camera changes while keeping trajectories and action targets fixed (Figure 5). We compare LAPA, UniVLA, FreshVLA, and FreshVLA without PAWKL. For each frozen model, the pretrained latent action encoder extracts latent actions from both clean and perturbed inputs, and the same MLP action regressor is trained on clean LIBERO trajectories and evaluated on perturbed inputs without adaptation, so that any degradation reflects perturbation sensitivity of the latent actions themselves. To isolate the model's intrinsic robustness to visual shifts from its base performance, we follow standard robustness evaluation protocols (Hendrycks & Dietterich, 2019) and report the Relative Robustness Score(RRS):

$$\mathrm{RRS} = \frac{\mathrm{MSE}_{\mathrm{clean}}}{\mathrm{MSE}_{\mathrm{perturbed}}}. \tag{16}$$

As shown in Figure 6(a), FreshVLA achieves the highest RRS across all perturbations, reaching 0.980, 0.980, and 0.933 for Background, Light, and Camera, respectively. Camera changes cause the largest performance gap, indicating that viewpoint variations are particularly disruptive to latent action representations. Removing PAWKL consistently degrades it, which confirms that FreshVLA effectively preserves action-relevant information, decoupling it from visual distractors.

B.1.2 DCT-Frequency Probe Analysis.

Our temporally hierarchical continuous latent action combines cross-timescale construction with DCT regularization to encourage relative spectral specialization. We examine whether the substage and execution streams exhibit the intended frequency preferences in decodable robot motion. Using frozen features extracted from the pretrained latent action encoder, we train four independent two-layer MLP probes from $\mathbf{s}$ or $\mathbf{e}$ to the low- or high-frequency coefficients of 6-D end-effector pose increments. Based on four LIBERO suites (Liu et al., 2023), we uniformly sample 36 frames, standardize the resulting 35 increments using training-set statistics, and apply an orthonormal temporal DCT (Ahmed et al., 1974; Pertsch et al., 2025; Huang et al., 2026). We assign bins 0–5 to the low band and 6–34 to the high band, using a fixed suite-stratified $70/15/15$ train/validation/test split.

For suite q and band b, we report the mean-baseline-normalized out-of-sample predictive R^2 (Hawinkel et al., 2024),

Table 7: Ablation study of the number of substages on LIBERO.

Substage Number	Long	Goal	Object	Spatial	Avg.
3	90.0	97.3	97.3	93.7	94.6
6	**98.0**	96.3	99.7	98.7	**98.2**
12	94.3	98.3	**100.0**	**99.3**	98.0

$$R^2_{\text{pred}}(q,b) \equiv G_{q,b} = 1 - \frac{\text{MSE}^{(q,b)}_{\text{probe}}}{\text{MSE}^{(q,b)}_{\text{mean}}}, \qquad \bar{R}^2_{\text{pred}}(b) = \frac{1}{4}\sum_{q=1}^{4} R^2_{\text{pred}}(q,b), \tag{17}$$

where the baseline uses the suite-specific mean coefficient vector and the macro score weights the four suites equally, avoiding scale- and suite-size-driven comparisons.

As shown in Figure 6(b), under the MLP probe, the substage stream achieves higher low-frequency predictive R^2 than the execution stream (0.315 vs. 0.284), whereas the execution stream achieves higher high-frequency predictive R^2 (0.053 vs. 0.001). Both streams remain more predictive in the low band, indicating shared access to slowly varying motion rather than an exclusive frequency partition. Their contrasting scores suggest relative spectral specialization: the substage stream exhibits a stronger low-frequency preference, while the execution stream retains more decodable high-frequency transition information. Overall, this pattern is consistent with the intended temporal organization encouraged by cross-timescale construction and DCT regularization.

B.2 Effect of Temporal Scale

The temporal granularity of hierarchical latent actions can significantly affect their representation quality. We therefore evaluate VLA performance with different numbers of substages under a fixed 36-frame input window. As the number of substages increases, the temporal span represented by the substage-scale latent action becomes shorter. As depicted in Table 7, using 6 or 12 substages outperforms using 3, with 6 substages achieving the highest average success rate of 98.2% and the best LIBERO-Long performance of 98.0%. This suggests that a suitable temporal granularity improves the modeling of long-horizon task dynamics.

C Real-World Generalization Analysis

To further assess the generalization capability of FreshVLA in real-world scenarios, we conduct qualitative evaluations under various out-of-distribution visual and physical shifts. As depicted in Figure 8, we introduce multiple compounding perturbations to the original tabletop setting for the **Organize Bowls** task. These testing conditions include:

- **Object Variation**: Altering the color, shape, and semantic identity of the manipulated objects.
- **Background Variation**: Introducing a large white mat to change the background texture and layout compared to the original green tabletop.
- **Position Variation**: Shifting the initial locations of the target objects and the receptacle.
- **Lighting Variation**: Reducing the environmental illumination.

These qualitative observations are consistent with our quantitative results on the simulated LIBERO-PLUS benchmark. The rollouts show that FreshVLA completes the Organize Bowls task despite substantial visual distractors, even when multiple visual perturbations are combined, suggesting robustness to real-world visual variation.

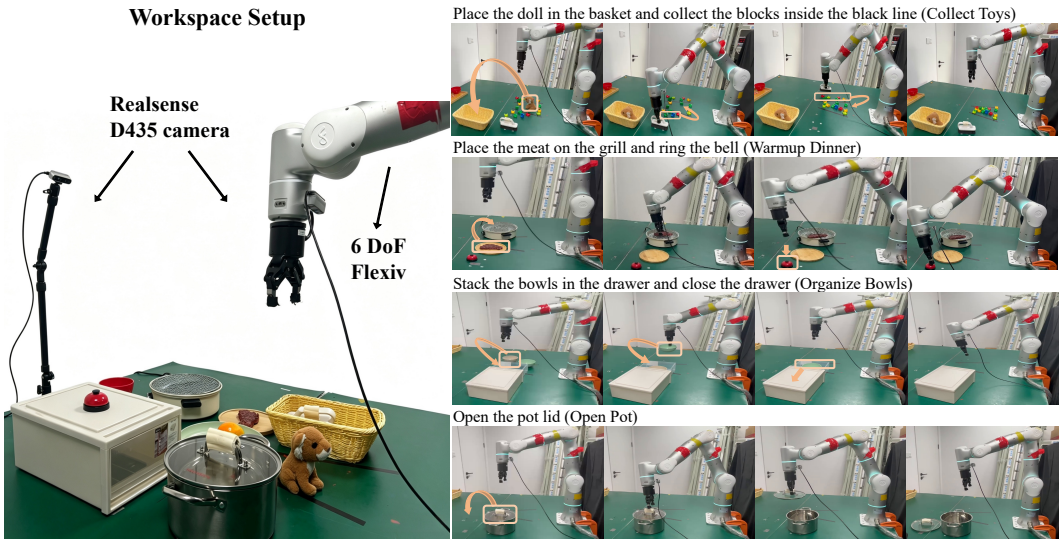

Figure 7: Real-world FLEXIV setup and representative manipulation sequences. Left: the 6-DoF Flexiv arm with a RealSense D435 camera. Right: representative rollouts for Collect Toys, Warmup Dinner, Organize Bowls, and Open Pot.

Table 8: Detailed provenance of the 65 repositories used from RoboCOIN.

Repository Name	Traj.	License	Repository Name	Traj.	License
AI2_Alphabot_2_pour_mung_beans	213	Apache-2.0	Airbot_MMK2_storage_stationery_xylophone	47	Apache-2.0
AI2_Alphabot_2_sort_personal_care_items	496	Apache-2.0	Airbot_MMK2_turn_page	14	Apache-2.0
AI2_Alphabot_2_store_toys_4	489	Apache-2.0	G1edu-u3_pick_apple_a	39	Apache-2.0
AI2_Alphabot_2_tidy_countertop	406	Apache-2.0	G1edu-u3_pick_cup_a	23	Apache-2.0
AIRBOT_MMK2_bowl_storage_pepper	9	Apache-2.0	G1edu-u3_pick_empty_bottle_ab	11	Apache-2.0
AIRBOT_MMK2_building_block_storage	41	Apache-2.0	G1edu-u3_pick_up_the_toy_ai	10	Apache-2.0
AIRBOT_MMK2_organize_and_place_books	17	Apache-2.0	G1edu-u3_place_bottle_c	28	Apache-2.0
AIRBOT_MMK2_take_down_paper_cup	1	Apache-2.0	G1edu-u3_place_metal_bowl_ae	15	Apache-2.0
Agilex_Cobot_Magic_basket_storage_peach	986	Apache-2.0	G1edu-u3_place_plastic_bowl_ag	7	Apache-2.0
Agilex_Cobot_Magic_connect_block	238	Apache-2.0	G1edu-u3_pullBowl_storage_bread_unordered_a	188	Apache-2.0
Agilex_Cobot_Magic_erase_board_left_side	101	Apache-2.0	G1edu-u3_put_the_lemon_af	2	Apache-2.0
Agilex_Cobot_Magic_fold_towel_yellow_tray	49	Apache-2.0	G1edu-u3_tray_storage_lemon_a	3	Apache-2.0
Agilex_Cobot_Magic_move_object_beige_tablecloth	199	Apache-2.0	Galaxea_R1_Lite_classify_object_green_tablecloth	224	Apache-2.0
Agilex_Cobot_Magic_pour_drink_bottle_cup_cup	200	Apache-2.0	Galaxea_R1_Lite_mix_red_yellow_large_test_tube	50	Apache-2.0
Agilex_Cobot_Magic_stir_coffee	495	Apache-2.0	Galaxea_R1_Lite_pour_solid	39	Apache-2.0
Agilex_Cobot_Magic_storage_bread_basket	89	Apache-2.0	Galaxea_R1_Lite_storage_object_brown_basket	15	Apache-2.0
Agilex_Cobot_Magic_storage_orange_basket_right	91	Apache-2.0	Galbot_G1_clear_sink_1	200	Apache-2.0
Cobot_Magic_cap_the_pen_a	55	Apache-2.0	Leju_Kuavo_4_sensor_card_swiping_11	600	Apache-2.0
Cobot_Magic_classification_of_fruits_and_vegetables	301	Apache-2.0	Leju_Kuavo_4_sensor_card_swiping_7	595	Apache-2.0
Cobot_Magic_close_book	272	Apache-2.0	R1_Lite_move_the_position_of_the_long_bread	31	Apache-2.0
Cobot_Magic_close_button	4	Apache-2.0	R1_Lite_open_and_close_the_freezer_door	93	Apache-2.0
Cobot_Magic_move_the_ball	100	Apache-2.0	R1_Lite_open_the_food_pan	94	Apache-2.0
Cobot_Magic_move_the_ball_and_the_cube_block	98	Apache-2.0	R1_Lite_opening_and_closing_aalcony_sliding_doors	101	Apache-2.0
Cobot_Magic_pour_drink	1613	Apache-2.0	R1_Lite_pick_up_and_store_items	108	Apache-2.0
Cobot_Magic_pull_zipper	1672	Apache-2.0	RMC-AIDA-L_organise_the_document_bag	480	Apache-2.0
Cobot_Magic_turn_on_the_desk_lamp	444	Apache-2.0	RMC-AIDA-L_pull_open_bag	1374	Apache-2.0
Airbot_MMK2_move_block	2	Apache-2.0	RMC-AIDA-L_stir_coffee	767	Apache-2.0
Airbot_MMK2_move_book_front	18	Apache-2.0	leju_robot_hotel_services_aa	90	Apache-2.0
Airbot_MMK2_move_cake	70	Apache-2.0	leju_robot_hotel_services_h	54	Apache-2.0
Airbot_MMK2_push_away_book	31	Apache-2.0	leju_robot_moving_parts_b	490	Apache-2.0
Airbot_MMK2_push_toy_car	5	Apache-2.0	leju_robot_moving_parts_f	489	Apache-2.0
Airbot_MMK2_storage_block_tape_measure	9	Apache-2.0	leju_robot_moving_parts_n	492	Apache-2.0
Airbot_MMK2_storage_spoon	51	Apache-2.0			

Table 9: Detailed provenance of the 16 original release subsets used from RoboMIND 2.0.

Repository Name	Training Group	Traj.	License	Repository Name	Training Group	Traj.	License
RoboMIND2.0-Agilex	agilex	69	Apache-2.0	RoboMIND2.0-Franka-Part-5	franka	16	Apache-2.0
RoboMIND2.0-Agilex-mobile	agilex_mobile	58	Apache-2.0	RoboMIND2.0-Franka-sim	franka_sim	120	Apache-2.0
RoboMIND2.0-Ark	ark	55	Apache-2.0	RoboMIND2.0-Tianyi	tianyi	173	Apache-2.0
RoboMIND2.0-Ark-mobile	ark_mobile	56	Apache-2.0	RoboMIND2.0-Tianyi-mobile	tianyi_mobile	253	Apache-2.0
RoboMIND2.0-Franka-Part-1	franka	15	Apache-2.0	RoboMIND2.0-Tienkung	tienkung	121	Apache-2.0
RoboMIND2.0-Franka-Part-2	franka	23	Apache-2.0	RoboMIND2.0-Tienkung-sim	tienkung_sim	1324	Apache-2.0
RoboMIND2.0-Franka-Part-3	franka	20	Apache-2.0	RoboMIND2.0-UR5	ur5	143	Apache-2.0
RoboMIND2.0-Franka-Part-4	franka	13	Apache-2.0	RoboMIND2.0-UR5-Dex	ur5_dex	162	Apache-2.0



Figure 8: Real-world generalization evaluation on the Organize Bowls task. The sequences illustrate FreshVLA successfully completing the multi-stage manipulation under compounding out-of-distribution perturbations, including variations in object appearance, background texture, object position, and environmental lighting.