FreeAct: Demonstration-Free Robot Adaptation via Action-Grounded Generated Videos

Zhe-Han Mo **Jia-Ning Li** **Jiaxi Song** **Meng-Hao Guo** **Shi-Min Hu***
BNRist, Department of Computer Science and Technology, Tsinghua University, China
{mozh23,ljn24,songjx22}@mails.tsinghua.edu.cn
{gmh,shimin}@tsinghua.edu.cn

Abstract

Adapting pretrained Vision-Language-Action (VLA) policies to new deployment environments typically requires collecting expert demonstrations in each target domain, which is costly and difficult to scale. In this paper, we present **FreeAct**, a framework for *demonstration-free robot adaptation* that converts generated videos into physically grounded supervision. A key challenge is that generated manipulation videos, while visually plausible, often contain embodiment artifacts and kinematic inconsistencies that make direct inverse-dynamics labeling unreliable. FreeAct addresses this challenge by learning a discrete latent action space from large-scale multi-lab robot videos and aligning it with camera-frame end-effector SE(3) motion and gripper-state changes. The resulting latent action tokenizer captures task-relevant motion while reducing reliance on embodiment-specific visual artifacts, enabling generated target-domain videos to be pseudo-labeled with reliable latent action tokens. These tokens are first injected into VLA pretraining and then used, together with source-domain action supervision, to adapt the policy to unseen environments. Empirically, FreeAct improves success from 2% to 57% under target-domain shift using generated videos alone, and further to 74% through self-evolution guided by a multimodal large language model (MLLM) on the policy's own rollouts, without expert demonstrations. Further analysis shows that the learned latent action space forms a kinematically structured and transferable representation, establishing generated videos as a scalable source of supervision for robot manipulation.

1 Introduction

Building robots that can follow open-ended instructions and operate reliably across diverse, unseen environments is a long-standing goal in robot learning [Brohan et al., 2023, Zitkovich et al., 2023, Guo et al., 2026a]. VLA models have made substantial progress toward this goal: pretrained on large-scale robot datasets and tuned on target-environment demonstrations, a single policy can perform diverse manipulation tasks across objects, scenes, and embodiments [Black et al., 2025a, Gemini Robotics Team et al., 2025, NVIDIA et al., 2025b]. Yet the last step, tuning on target-environment demonstrations, remains a key bottleneck for deployment. Policies trained in one environment overfit to its distribution and degrade sharply under changes in lighting, texture, camera viewpoint, distractors, object instances, or robot appearance [Yu et al., 2023, Fei et al., 2026], while collecting demonstrations for every new environment is costly, slow, and difficult to scale [O'Neill et al., 2024]. This motivates a practically important setting we term *demonstration-free robot adaptation*: adapting a pretrained policy to a new environment without any target-environment expert demonstrations.

Recent advances in video generation [Zheng and Huang, 2026] offer a compelling path toward this goal. Trained on internet-scale corpora, foundation video generators can synthesize visually realistic

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

manipulation videos in a zero-shot manner from language instructions and reference images, naturally covering the broad range of appearance variation, scene diversity, and long-tail conditions [Brooks et al., 2024, NVIDIA et al., 2025a]. In principle, such generated videos could serve as deployment-specific experience for adapting pretrained VLA policies, eliminating the need for target-environment teleoperation. The central challenge is recovering executable robot actions from these videos. A natural approach is to synthesize manipulation videos and then apply an Inverse Dynamics Model (IDM) [Du et al., 2023, Tian et al., 2025]. However, we empirically find this pipeline to be unreliable when the video generator is used without domain-specific finetuning. Generated videos look plausible, but often contain kinematic inconsistencies such as non-rigid arm deformations. These artifacts fall outside the IDM's training distribution and corrupt the inferred actions [Liu et al., 2026]. Tuning the generator on robot-specific data could suppress these artifacts, but is computationally expensive and risks eroding its broad visual coverage and physical priors [Jang et al., 2025]. We therefore ask: *can generated videos be converted into physically grounded action supervision for robot adaptation, without finetuning the generator?*

To this end, we propose **FreeAct**, a three-stage framework that learns physically grounded latent action representations and uses them to adapt pretrained VLA policies from generated videos. In the first stage, we train a latent action model (LAM) on large-scale robot videos with a VQ-VAE objective [van den Oord et al., 2017], augmented by an auxiliary Action Decoder that supervises the latent bottleneck to predict camera-frame end-effector motion. This grounding pressures the latent space to encode kinematics rather than embodiment appearance, yielding a representation that remains reliable on generated videos. In the second stage, the resulting discrete tokens serve as prediction targets for VLA pretraining, injecting a physically grounded motion abstraction into the policy backbone without requiring action labels. In the third stage, we attach a continuous action head and jointly finetune on action-labeled source data and generated target-environment videos pseudo-labeled by the stage-1 model, with a projected gradient update to reconcile the two data streams. Together, this pipeline converts generated videos into action-grounded supervision for demonstration-free policy adaptation.

The practical value of FreeAct lies in its ability to close the full adaptation loop without expert intervention. Generated target-environment videos alone lift a pretrained VLA from 2% to 57% success under substantial domain shift. The adapted policy then refines itself: a handful of its own rollouts, selected by a multimodal LLM judge, raise success to 74%. The two phases are complementary: generated videos bootstrap broad task competence, while self-refinement corrects residual errors such as depth ambiguities that generated supervision alone cannot resolve. A suite of diagnostics reveals a consistent pattern: reconstruction-only baselines yield fragmented, appearance-driven latent spaces, whereas our physically grounded model induces a continuous action manifold whose neighborhoods correspond to similar end-effector motions. This structural alignment is what enables FreeAct to turn generated videos from visually plausible trajectories into actionable supervision for robot adaptation.

Our contributions are threefold:

1. We propose FreeAct, a three-stage framework that grounds discrete latent actions in camera-frame end-effector kinematics, transfers this motion abstraction into a pretrained VLA, and adapts the policy from action-grounded generated videos to a target environment without expert demonstrations.
2. We introduce a set of diagnostics for evaluating latent action spaces and show that physical grounding produces a continuous, linearly decodable kinematic manifold, in contrast to the fragmented, appearance-driven representations learned by reconstruction-only baselines.
3. We show that FreeAct enables demonstration-free adaptation under aggressive domain shift, lifting a pretrained VLA from 2% to 57% success with generated videos alone, and to 74% through self-evolution on a handful of the policy's own rollouts.

2 Related Work

Vision-Language-Action models VLA models adapt pretrained MLLMs to robot control, predicting action tokens or continuous action chunks from visual observations and language instructions [Zitkovich et al., 2023, Kim et al., 2025, Black et al., 2025b, NVIDIA et al., 2025b]. Recent

generalist policies scale this paradigm with large robot datasets, cross-embodiment mixtures, expressive action decoders, and richer conditioning signals such as memory, subtask prompts, subgoal images, and embodiment-aware prompts [O'Neill et al., 2024, Ghosh et al., 2024, Zheng et al., 2026, Physical Intelligence et al., 2026]. Despite their growing generality, deploying these policies to a new environment typically still requires target-domain robot data, whether through demonstrations, post-training, or carefully curated cross-embodiment pretraining. We target the more practical regime in which no such target-domain robot data is provided.

Video generation for robot learning Video generators provide spatio-temporal priors increasingly exploited for robot learning. One line executes generated videos at inference time, recovering actions with an IDM or a goal-conditioned policy [Du et al., 2023, 2024] or by tracking a tool or object across frames [Liang et al., 2025, Patel et al., 2026]; Gen2Act [Bharadhwaj et al., 2025] instead conditions a policy on generated human videos while learning actions from robot demonstrations. Another line trains a world model, typically from a pretrained generator, and either labels or evaluates its rollouts [Jang et al., 2025, Guo et al., 2026b, Gao et al., 2026] or predicts actions jointly with future visual states [Zhu et al., 2025a, Bi et al., 2026, Kim et al., 2026, Li et al., 2026, Ye et al., 2026, Lyu et al., 2026]. We instead keep the generator off-the-shelf and convert its videos into action supervision through a physically grounded LAM.

Latent action learning Another line of work learns latent actions from action-free videos by encoding inter-frame transitions through a discrete bottleneck and using the resulting codes for control or policy pretraining [Bruce et al., 2024, Schmidt and Jiang, 2024, Cui et al., 2024, Ye et al., 2025, Bu et al., 2025, Chen et al., 2026]. Such objectives primarily constrain visual predictability, leaving the relationship between latent codes and executable robot motion only indirectly specified. Liu et al. [2025] align video-derived latent actions with environment actions through online interaction; we instead retain the action-free latent objective and ground it offline with end-effector motion from a calibrated subset of a multi-lab robot corpus.

3 Method

Problem setup We study *demonstration-free robot adaptation*: given a VLA policy pretrained on source environments, our goal is to adapt it to a target environment without any target-environment expert demonstrations. Let

$$\mathcal{D}_{\mathrm{src}} = \{(o_t,\, o_{t+\Delta},\, \ell_t,\, a_t)\}$$

denote action-labeled robot data from source environments, where o_t and $o_{t+\Delta}$ are RGB observations separated by a fixed temporal horizon Δ, ℓ_t is a language instruction, and

$$a_t = [\Delta p_t,\, \Delta r_t,\, \Delta g_t] \in \mathbb{R}^7$$

is the camera-frame end-effector action comprising 3D translation $\Delta p_t \in \mathbb{R}^3$, 3D axis-angle rotation $\Delta r_t \in \mathbb{R}^3$, and scalar gripper-state change $\Delta g_t \in \mathbb{R}$. In addition, we have access to videos synthesized by an off-the-shelf video generator for the target domain,

$$\mathcal{D}_{\mathrm{gen}} = \{(o^g_t,\, o^g_{t+\Delta},\, \ell^g_t)\},$$

which provide visual trajectories under target-domain appearance but carry no real robot actions. In the *self-evolution* extension, a small number of the adapted policy's own target-environment rollouts, selected by an automated quality signal, form $\mathcal{D}_{\mathrm{self}}$.

FreeAct addresses this setting through three stages: Stage 1 learns a *physically grounded LAM* from large-scale multi-lab robot videos, using a calibrated subset to anchor the latent space to camera-frame kinematics. Stage 2 uses the resulting discrete latent action tokens as supervision for VLA pretraining, injecting motion abstraction into the policy backbone without requiring dense action labels. Stage 3 adapts the pretrained VLA to the target environment by jointly optimizing over source action-labeled data and pseudo-labeled generated videos. At test time, the adapted policy maps a target-environment observation and language instruction directly to a continuous robot action.

3.1 Physically grounded latent action learning

Stage 1 builds on the standard latent action model [van den Oord et al., 2017, Schmidt and Jiang, 2024] to learn a discrete latent representation of motion from frame pairs and ground it in executable

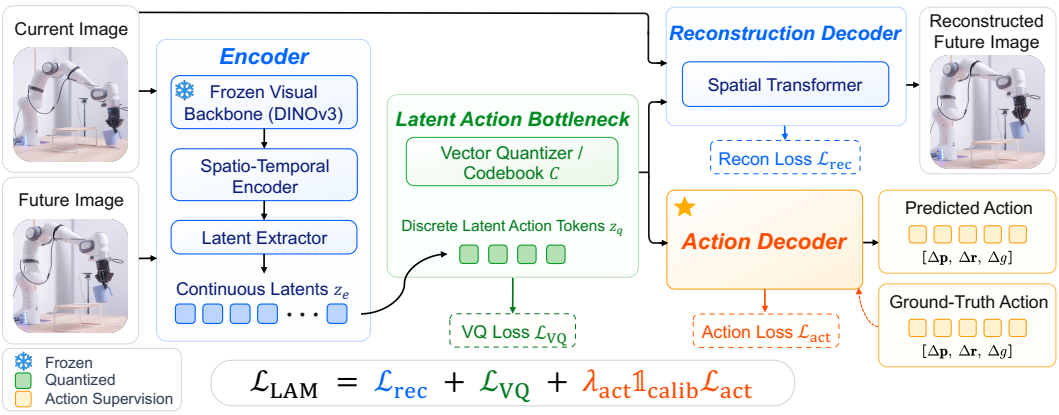


Figure 1: **Stage-1 latent action model.** A frozen DINOv3 encoder maps a frame pair to continuous latents z_e, vector-quantized into discrete tokens z_q against codebook $\mathcal{C}$. A reconstruction decoder predicts the future frame ($\mathcal{L}_{\text{rec}}$, $\mathcal{L}_{\text{VQ}}$). Our key addition ($\star$) is an Action Decoder that maps z_q to the camera-frame end-effector action and is supervised on the calibrated subset ($\mathcal{L}_{\text{act}}$), grounding the codebook in executable kinematics.

robot kinematics. A spatio-temporal encoder Q_ϕ maps a frame pair $(o_t, o_{t+\Delta})$ into a sequence of K continuous latents, which are vector-quantized against a learnable codebook $\mathcal{C}$ to yield discrete tokens

$$z_t = [z_t^{(1)}, \ldots, z_t^{(K)}], \qquad z_t^{(k)} \in \{1, \ldots, |\mathcal{C}|\},$$

and a decoder D_{rec} reconstructs the future frame from z_t and o_t. We use a frozen DINOv3 ViT [Siméoni et al., 2026] as the visual backbone and a transformer encoder–decoder [Xu et al., 2022, Bao et al., 2025] for the spatio-temporal and reconstruction modules; full architectural specifications are deferred to Appendix A. Figure 1 illustrates the resulting pipeline, including our auxiliary Action Decoder branch.

Physical grounding Trained with reconstruction alone, this template offers no pressure on *what* the latent space encodes. The bottleneck is free to organize itself around appearance, texture, or embodiment-specific cues, and prior LAMs exhibit exactly this failure: fragmented, appearance-driven codebooks whose neighborhoods carry little kinematic meaning (Sec. 4.2). We address this with an auxiliary Action Decoder

$$\hat{a}_t = D_{\text{act}}(z_t),$$

a lightweight multi-layer perceptron (MLP) that maps the quantized latents directly to a predicted camera-frame end-effector action. The branch is supervised only on the calibrated subset of $\mathcal{D}_{\text{src}}$ for which camera–robot extrinsics are available.

The design decision is to use the *camera frame* rather than a shared world frame as the supervision target. Camera extrinsics vary substantially across laboratories and episodes, so a model that shortcuts action prediction via embodiment appearance or scene layout cannot generalize across episodes. The latent space is instead pressured to encode end-effector motion relative to the camera in a manner that generalizes across viewpoints. This inductive bias remains valid on generated videos: while full-arm geometry is frequently hallucinated, end-effector motion relative to the scene tends to stay faithful to the task.

Training objective The stage-1 loss combines reconstruction, VQ commitment, and the action supervision above:

$$\mathcal{L}_{\text{LAM}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{VQ}} + \lambda_{\text{act}}\, \mathbb{1}_{\text{calib}}\, \mathcal{L}_{\text{act}} \tag{1}$$

where $\mathbb{1}_{\text{calib}}$ gates the action loss to the calibrated subset. The 7D action is decomposed into translation (p), rotation (r), and gripper (g) components, balanced by homoscedastic uncertainty weighting [Kendall et al., 2018]:

$$\mathcal{L}_{\text{act}} = \sum_{m \in \{p,r,g\}} \left(e^{-s_m}\, \text{SmoothL1}\left(\hat{a}_t^{(m)},\, a_t^{(m)}\right) + s_m \right) \tag{2}$$

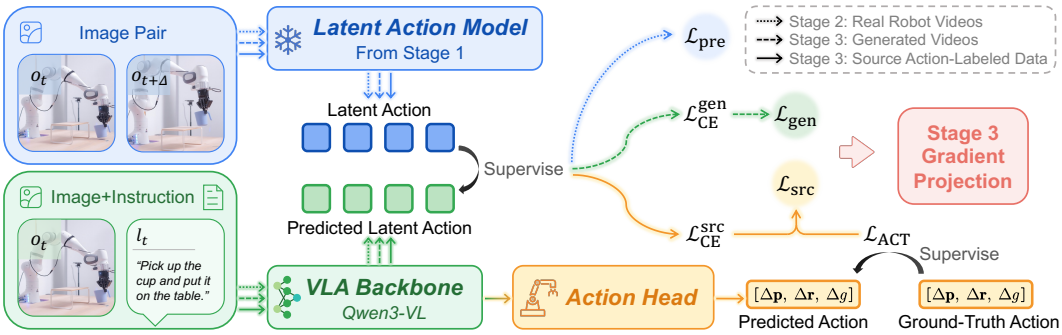


Figure 2: **Stages 2 and 3 of FreeAct.** The frozen stage-1 latent action model (top-left) tokenizes a frame pair into a discrete sequence, which the VLA backbone is trained to autoregressively predict from the current image and instruction ℓ_t. Stage 2 (*dotted*) pretrains the backbone on real robot videos with cross-entropy loss $\mathcal{L}_{\text{pre}}$. Stage 3 jointly trains two streams: source action-labeled data (*solid*), supervising both the latent tokens and a continuous action head ($\mathcal{L}_{\text{src}}$); and pseudo-labeled generated target-environment videos (*dashed*), supervising only the latent tokens ($\mathcal{L}_{\text{gen}}$). The two stage-3 streams are combined through the gradient projection update of Sec. 3.3.

with learnable log-variances $s_m = \log \sigma_m^2$ that absorb relative modality scales during training.

After stage 1, the model serves two roles: it provides a physically grounded latent action space whose neighborhoods reflect executable motion rather than visual similarity, and it acts as a fixed tokenizer that maps real and generated frame pairs to discrete action sequences for downstream stages.

3.2 Latent action-guided VLA pretraining

The second stage transfers the motion abstraction into the VLA backbone (Figure 2). We augment the backbone's text tokenizer vocabulary with $|\mathcal{C}|$ special action tokens $\{\texttt{<la}_0\texttt{>}, \ldots, \texttt{<la}_{|\mathcal{C}|-1}\texttt{>}\}$, one per codebook entry, and cast latent action prediction as next-token prediction within the backbone's chat template:

$$\begin{aligned} \texttt{User:} \quad & \texttt{<image>}\, \ell_t \\ \texttt{Assistant:} \quad & \texttt{<la}_{z_t^{(1)}}\texttt{>} \cdots \texttt{<la}_{z_t^{(K)}}\texttt{>}. \end{aligned}$$

For each training sample $(o_t, o_{t+\Delta}, \ell_t)$, the frozen stage-1 tokenizer provides the target sequence $z_t = Q_\phi(o_t, o_{t+\Delta})$. The VLA backbone is then trained to predict it autoregressively:

$$\mathcal{L}_{\text{pre}} = -\sum_{k=1}^{K} \log p_\theta\left(z_t^{(k)} \mid o_t,\, \ell_t,\, z_t^{(<k)}\right). \tag{3}$$

The visual encoder is kept frozen throughout; only the vision–language projector and language model are updated. This objective scales beyond densely annotated trajectories, since latent action tokens can be extracted from any unlabeled video. It exposes the backbone to a physically grounded motion representation before continuous control is introduced in stage 3.

3.3 Deployment adaptation via generated videos

In the final stage, we adapt the pretrained VLA to the target environment by jointly optimizing over action-labeled source data and generated target-environment videos (right side of Figure 2).

Continuous action head We attach a continuous action head H_ψ that decodes the hidden states at the latent-action token positions into a continuous robot action, $\hat{a}_t = H_\psi(h_t)$. H_ψ is built on multihead attention pooling (MAP) [Lee et al., 2019]: it pools the latent-token hidden states into a compact action embedding and projects it to the action space. The head is trained from scratch in stage 3, while the backbone inherits the representation from stage 2.

Joint finetuning For source samples $(o_t, o_{t+\Delta}, \ell_t, a_t) \in \mathcal{D}_{\text{src}}$, we supervise both the latent token sequence and the continuous action:

$$\mathcal{L}_{\text{src}} = \mathcal{L}_{\text{CE}}^{\text{src}} + \lambda_a \mathcal{L}_{\text{ACT}}, \qquad \mathcal{L}_{\text{ACT}} = \|\hat{a}_t - a_t\|_1. \tag{4}$$

For generated video samples $(o_t^g, o_{t+\Delta}^g, \ell_t^g) \in \mathcal{D}_{\text{gen}}$, no real actions are available; we pseudo-label each sample with the frozen stage-1 tokenizer, $\tilde{z}_t^g = Q_\phi(o_t^g, o_{t+\Delta}^g)$, and supervise only the latent token prediction:

$$\mathcal{L}_{\text{gen}} = \lambda_v \mathcal{L}_{\text{CE}}^{\text{gen}}. \tag{5}$$

The two streams thus play asymmetric roles: source data anchors continuous control fidelity, while generated videos expose the backbone to how motion manifests under target-domain appearance, viewpoint, and object variation.

Gradient conflict mitigation This asymmetry makes naive joint optimization unstable: the two streams carry different supervision signals, one continuous and one token-level, and their gradients frequently conflict. When this happens, the generated gradient erodes the precise control signal carried by action labels. To mitigate this, we adopt a gradient projection update tailored to our two-stream setting [Yu et al., 2020]. Let g_{src} and g_{gen} denote the per-parameter gradients from each stream. Whenever the gradients conflict, $g_{\text{src}}^\top g_{\text{gen}} < 0$, we project away the conflicting component

$$\tilde{g}_{\text{gen}} \leftarrow g_{\text{gen}} - \frac{g_{\text{src}}^\top g_{\text{gen}}}{\|g_{\text{src}}\|^2} g_{\text{src}} \tag{6}$$

and update with $g_{\text{src}} + \tilde{g}_{\text{gen}}$. This preserves the continuous control signal from source data while keeping the target-domain visual gradient alive on parameters where the two streams agree.

4 Experiments

We evaluate FreeAct along the full pipeline from latent action learning to deployment adaptation. Our experiments are organized around three research questions:

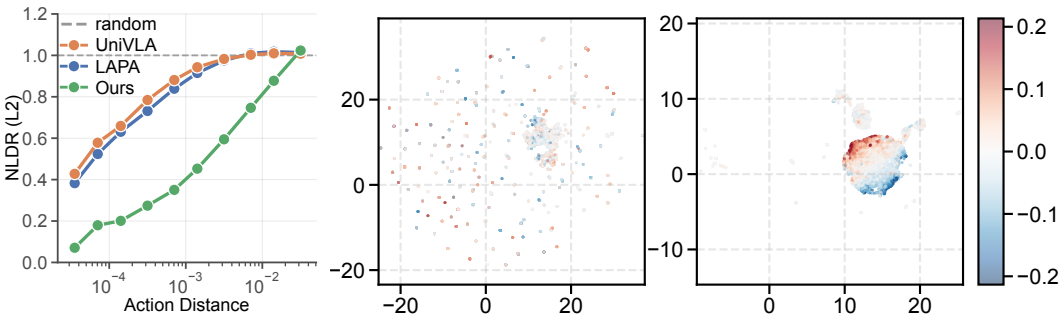
- **RQ1** Does physically grounded supervision yield a kinematically structured latent action space, relative to purely reconstruction-based baselines? (Sec. 4.2)
- **RQ2** Can generated videos alone adapt a pretrained VLA policy to unseen target environments without any real demonstrations, and does our grounded LAM provide more reliable pseudo-supervision than IDM-based and reconstruction-only alternatives? (Sec. 4.3)
- **RQ3** Can the adapted policy further improve through self-evolution on its own rollouts, without expert intervention? (Sec. 4.4)

4.1 Experimental setup

Datasets Stages 1 and 2 use DROID [Khazatsky et al., 2024] for latent action learning and VLA pretraining; the calibrated subset with camera extrinsics provides $\mathcal{L}_{\text{act}}$ supervision. Stage 3 draws source data from Liu et al. [2023] and evaluates on a custom robosuite [Zhu et al., 2025b] target environment with substantial visual and task shift. Target-domain videos are synthesized with Wan 2.6 [Wan et al., 2025] (20 rollouts per task). Full setup details are in Appendix B.

Baselines For the latent space evaluation (RQ1), we compare against LAPA [Ye et al., 2025] and UniVLA [Bu et al., 2025], two LAMs trained with reconstruction objectives alone. For end-to-end adaptation (RQ2), we compare against UniPi [Du et al., 2023], which recovers continuous actions from generated videos via an IDM, DreamGen [Jang et al., 2025], which finetunes the generator before IDM labeling, and the LAPA and UniVLA tokenizers swapped into our stage-2/3 pipeline.

Implementation details Stage 1 uses a frozen DINOv3 ViT-B/16 [Siméoni et al., 2026] as the visual backbone with a transformer encoder–decoder for the VQ-VAE. Stage 2 initializes from Qwen3-VL-4B [Bai et al., 2025], keeping the visual encoder frozen and finetuning the projector and language model. Stage 3 inherits the stage-2 checkpoint and jointly trains the continuous action head and backbone on source and generated data. Full hyperparameters and training schedules are in Appendix A.



(a) NLDR. (b) UMAP projection of test-set latents.

Figure 3: **Latent action space structure on the DROID held-out test split.** (a) NLDR vs. action-distance threshold τ; lower values indicate that physically similar actions are colocated in the latent space. (b) UMAP projection of test-set latents, colored by end-effector x-displacement in meters (left: UniVLA; right: ours). LAPA and additional axes are deferred to Appendix D.

4.2 RQ1: Physically grounded latent action space

We first evaluate the intrinsic quality of the latent action space learned in stage 1, asking whether physical action supervision produces a representation whose geometry reflects executable kinematics rather than visual texture. All metrics below are computed on a held-out DROID test split disjoint from our training trajectories. Baselines are LAPA [Ye et al., 2025] and UniVLA [Bu et al., 2025], using their official checkpoints; both were trained on broader datasets that include our test trajectories.

Topology preservation A well-structured latent space should bring physically similar actions closer than dissimilar ones. We quantify this with a *normalized latent-distance ratio* (NLDR): for an action-distance threshold τ, we compare the mean latent distance among sample pairs whose ground-truth actions are within τ to the mean distance over random pairs,

$$\mathrm{NLDR}(\tau) = \frac{\mathbb{E}_{(i,j)\in\mathcal{S}_\tau}\|z_i - z_j\|_2}{\mathbb{E}_{\text{random}}\|z_u - z_v\|_2}, \qquad \mathcal{S}_\tau = \{(i,j) : d_a(a_i, a_j) < \tau\}. \tag{7}$$

The action distance d_a sums the mean SmoothL1 distance over the three semantic components of the 7D action: translation (p), rotation (r), and gripper (g), giving each component equal weight regardless of its dimensionality. A value near 1 indicates no correlation between latent and physical distance; values near 0 indicate that physically similar actions form tight latent clusters. As shown in Figure 3a, our model reaches NLDR below 0.1 at the smallest τ and stays well below the baselines across the full range; LAPA and UniVLA improve only marginally over random as τ shrinks, with NLDR remaining around 0.4 even at the tightest threshold.

Manifold structure Figure 3b projects test-set latents into 2D [Yuan et al., 2021] via UMAP [McInnes et al., 2020], colored by ground-truth end-effector x-displacement. The UniVLA latent space disperses into a noisy central cluster in which colors are intermixed with no discernible gradient: the continuous action space is fractured into unrelated fragments, leaving no geometric reference along which a downstream policy can interpolate. Our model instead forms a dense manifold whose color varies smoothly from one side to the other, mapping the topology of physical motion onto the latent space. Adjacent latent vectors thus correspond to neighboring physical actions, the prerequisite for fine-grained control and smooth skill generalization in downstream stages.

Linear decodability A continuous manifold structure is only useful if downstream modules can read off action information from it. We probe this with ridge regression from frozen

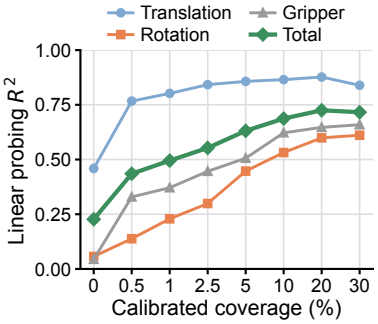
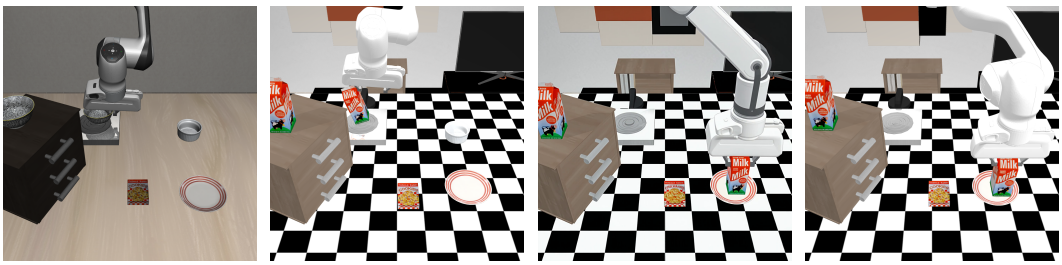


Figure 4: **Linear probing R^2 vs. action-supervision coverage.** Stage 1 is retrained with $\mathcal{L}_{\text{act}}$ restricted to nested subsets of calibrated trajectories.

Table 1: **Linear probing R^2 on the DROID held-out test set.** Reconstruction-only baselines fail on rotation and gripper dimensions; our model achieves high decodability across all components.

Method	LAPA		UniVLA		Ours	
	Train	Test	Train	Test	Train	Test
Total R^2	0.104	0.107	0.096	0.090	0.794	**0.761**
Translation R^2	0.223	0.226	0.201	0.189	0.899	**0.878**
Rotation R^2	0.013	0.014	0.016	0.013	0.711	**0.660**
Gripper R^2	0.022	0.029	0.019	0.021	0.727	**0.712**
Norm. MSE	0.896	0.990	0.904	1.008	0.206	**0.264**



(a) Source Env (b) Target Env (c) Generated Video (d) Rollout

Figure 5: **An example of the source–target shift, generated supervision, and resulting rollout (Task E).** (a) Source environment. (b) Target environment, exemplifying the visual and task changes considered in this paper; without adaptation, the source policy topples the milk carton with its pinch trajectory. (c) Frame from a Wan 2.6 generated rollout: the arm shows kinematic hallucinations, yet the gripper and end-effector remain faithful. (d) Successful FreeAct rollout after adaptation.

latents to the 7D ground-truth action, reporting the coefficient of determination R^2 (higher is better) on train and test splits. Table 1 shows that LAPA and UniVLA achieve $R^2 \approx 0.1$ overall, with rotation and gripper components effectively undecodable ($R^2 < 0.03$). Our model reaches $R^2 = 0.76$ overall, with translation, rotation, and gripper all linearly decodable in their own right. The latent space not only clusters by kinematics, it explicitly encodes them.

Supervision coverage Camera extrinsics are available for only $\sim$40% of DROID, so we vary action-supervision coverage to assess how much calibrated data grounding requires. We retrain stage 1 with the action loss restricted to nested subsets of calibrated trajectories, keeping all other data and settings fixed (Figure 4). Within this sweep, reducing coverage from 30% to 0% ($\lambda_{\text{act}} = 0$) lowers R^2 from 0.72 to 0.23, with rotation and gripper falling from 0.61/0.66 to 0.06/0.04. Performance levels off around 20% coverage; 1% already attains 69% of the plateau value, indicating that limited calibration can substantially improve action decodability.

Across these diagnostics, the picture is consistent: physically grounded supervision through the auxiliary Action Decoder is sufficient to convert a reconstruction-only codebook into a continuous, linearly decodable kinematic manifold. This structural property is what allows downstream stages to convert generated videos into reliable adaptation supervision (Sec. 4.3).

4.3 RQ2: Demonstration-free adaptation from generated videos

We now turn to the central question of this paper: can a pretrained VLA policy be adapted to a new environment using only generated videos, with zero expert demonstrations? We construct a target environment with changes in lighting, desk textures, robot link colors, initial joint configurations, task instructions, and most importantly the target manipulation skill itself (Figure 5a, b). Source tasks reduce to a *precision pinch* on a thin protrusion, exemplified by the rim of a bowl, where the policy only needs to align one finger with a narrow edge before closing. Target tasks instead require an *enclosing grasp* on a milk carton, an object never seen in the source distribution: the gripper must approach with both fingers straddling the full width of the body, at a different height, direction, and

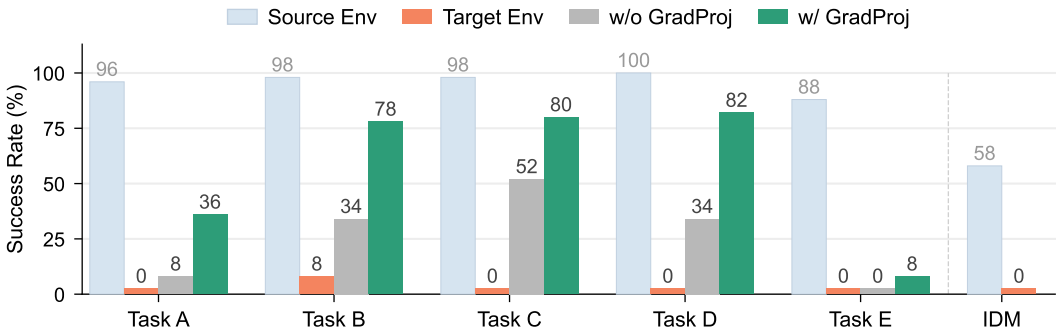


Figure 6: **Demonstration-free adaptation results on five target tasks.** *Source Env*: pretrained VLA on the source environment (sanity check). *Target Env*: same policy deployed on the target environment without adaptation. *w/ GradProj* (FreeAct): adaptation using only Wan 2.6 generated videos pseudo-labeled by our latent action model, with the gradient projection update of Sec. 3.3. *w/o GradProj*: the same setup with the projection removed. *IDM*: UniPi-style baseline, evaluated on real source videos (left, 58%) and on generated target videos (right, 0%). Each bar averages 50 closed-loop rollouts.

closing pattern. The two regimes thus exercise different aspects of low-level control, not merely different appearances.

Figure 6 reports closed-loop success across five target tasks, averaged over 50 trials each. The pretrained VLA, despite near-saturated source performance (96% on average), collapses to 1.6% when deployed directly on the target environment. Lacking any prior exposure to carton-shaped objects, the policy keeps executing its source-domain pinch trajectory, swinging one finger of the parallel gripper into the carton and toppling it (Figure 5b). FreeAct, supervised only by Wan 2.6 generated videos (Figure 5c), recovers an average success rate of 57% (Figure 5d), with four of five tasks exceeding 35% and three exceeding 75%.

A natural alternative is to recover actions directly from generated videos via IDM, as in UniPi. We evaluate it on two inputs separately. On real teleoperated source-environment videos the IDM is faithful, replaying its predictions with 58% success and confirming our reimplementation is sound. On Wan 2.6 generated videos the same IDM collapses to 0%: kinematic inconsistencies push the inputs outside its training distribution, and the recovered action sequences become physically incoherent. DreamGen's official pipeline, which uses a robot-finetuned video generator followed by IDM labeling, also yields 0% success. While the generated arm is hallucinated, the gripper and end-effector motion remain faithful to the task; our latent action model is designed to extract exactly this signal (Sec. 3.1). For comparison, we use the official reconstruction-only LAPA or UniVLA tokenizer in the same stage-2/3 pipeline. Source success remains high (87% and 89%), but target success reaches only 12% and 4% (Table 6), consistent with their lower action decodability in RQ1.

Removing the gradient projection step (w/o GradProj in Figure 6) drops the average success rate from 57% to 26%, with three of five tasks losing more than half of their success and Task E collapsing to 0%. This confirms the prediction of Sec. 3.3: a naive sum of the two losses lets the generated-video gradient cancel parts of the source-stream update, eroding the precise control signal carried by action labels. The projected update strips the conflicting component from the generated gradient, so source data continues to deliver clean control supervision while the generated stream still propagates target-domain visual information through the parameters where the two streams agree.

Task E remains the hardest case across all settings, including for FreeAct (8%). The cause traces to the target scene geometry: the milk carton sits almost directly beneath the gripper and slightly farther from the camera, weakening the perspective cues that disambiguate depth. Wan 2.6 interprets this as a vertical descent and generates videos in which the gripper drops straight down, ignoring the small forward displacement required for accurate grasping. The downstream policy is therefore upper-bounded by the fidelity of the generated supervision: when the generator misreads the scene geometry, no amount of action grounding can recover the missing kinematic content. This points toward generation-side improvements such as multi-view conditioning or depth priors [Xu et al., 2026] as a complementary axis to action grounding.

4.4 RQ3: Self-evolution from policy rollouts

The adapted policy from RQ2 recovers basic task competence from generated videos alone. We now ask whether it can further improve through its own experience. We deploy the stage-3 policy in the target environment, score each rollout with GPT-5.5 [OpenAI, 2026] on task completion and execution quality (prompt in Appendix C), and finetune on up to five of the highest-scoring rollouts per task, drawn from the 50 evaluation rollouts per task in Figure 6. Ground-truth success labels are not used for selection; a post hoc check confirms that all selected rollouts are successful.

Figure 7 reports success rates after one round of self-evolution. The average rises from 57% to 74%. Task E, previously capped at 8% due to depth ambiguity in the generated videos, jumps to 52%, confirming that even minimal real interaction can correct biases that generated supervision alone cannot resolve. Task D falls from 82% to 56% despite successful selected rollouts. Pooling data across tasks may introduce interference during finetuning, a possible explanation for this decline.

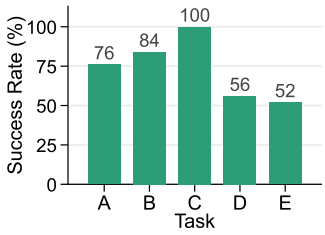


Figure 7: **Self-evolution from rollouts.** Success rates after one round of self-evolution. Each bar averages 50 closed-loop rollouts.

No target-domain expert demonstration is introduced at any point: stage 3 adapts from generated videos, and self-evolution draws on the policy's own rollouts as judged by an MLLM. This establishes a complete adaptation loop for deployment to novel environments, in which generated videos bootstrap competence and autonomous refinement lifts performance to practical levels.

5 Conclusion

We presented FreeAct, a demonstration-free adaptation framework that anchors a discrete latent action space to camera-frame end-effector kinematics, enabling reliable pseudo-labeling of generated videos despite their kinematic artifacts. Generated videos alone lift a pretrained VLA from 2% to 57% under aggressive domain shift; the adapted policy then self-evolves on its own rollouts to 74%. These results establish a complete adaptation loop that requires no target-domain expert demonstrations.

Limitations and future work Our evaluation is confined to simulation, a single embodiment, and single-arm tabletop manipulation; real robots, cross-embodiment transfer, and bimanual or dexterous settings with richer contact dynamics are natural next steps. Adaptation is bounded by generator fidelity (Task E), which multi-view conditioning or depth priors could improve. Pooled self-evolution can also reduce performance on individual tasks (Task D); per-task balancing is a possible remedy.

6 Acknowledgements

This work was supported by Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM101), the National Natural Science Foundation of China (project No. 62220106003, 62495061, 623B2057).

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report, 2025. URL `https://arxiv.org/abs/2511.21631`.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Text to image generation with bidirectional multiway transformers. *Computational Visual Media*, 11(2):405–422, 2025. doi: 10.26599/CVM.2025.9450377.

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2Act: Human video generation in novel scenarios enables generalizable robot manipulation. In Joseph Lim, Shuran Song, and Hae-Won Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 3936–3951. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/bharadhwaj25a.html.

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)*, pages 35101–35113, June 2026.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In Joseph Lim, Shuran Song, and Hae-Won Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 17–40. PMLR, 27–30 Sep 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025b. doi: 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.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators, 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. 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 Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 4603–4623. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

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.

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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=y5CaJb17Fn.

Zichen Jeff Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. DynaMo: In-domain dynamics pretraining for visuo-motor control. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 33933–33961. Curran Associates, Inc., 2024. doi: 10.52202/079017-1069. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/3b8db54b629e00537b59cbc6612026d7-Paper-Conference.pdf.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 9156–9172. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1d5b9233ad716a43be5c0d3023cb82d0-Paper-Conference.pdf.

Yilun Du, Sherry Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Pack Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9pKtcJcMP3.

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)*, pages 38574–38583, June 2026.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Naveed Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, Jiannan Xiang, Yuqi Xie, Ruijie Zheng, Dantong Niu, Pooya Jannaty, Jinwei Gu, Jun Zhang, Jitendra Malik, Pieter Abbeel, Ming-Yu Liu, Yuke Zhu, Joel Jang, and Linxi Fan. DreamDojo: A generalist robot world model from large-scale human videos. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=FuvU7PTyED.

Gemini Robotics Team, Abbas Abdolmaleki, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Ashwin Balakrishna, Nathan Batchelor, Alex Bewley, Jeff Bingham, Michael Bloesch, Konstantinos Bousmalis, Philemon Brakel, Anthony Brohan, Thomas Buschmann, Arunkumar Byravan, Serkan Cabi, Ken Caluwaerts, Federico Casarini, Christine Chan, Oscar Chang, London Chappellet-Volpini, Jose Enrique Chen, Xi Chen, Hao-Tien Lewis Chiang, Krzysztof Choromanski, Adrian Collister, David B. D'Ambrosio, Sudeep Dasari, Todor Davchev, Meet Kirankumar Dave, Coline Devin, Norman Di Palo, Tianli Ding, Carl Doersch, Adil Dostmohamed, Yilun Du, Debidatta Dwibedi, Sathish Thoppay Egambaram, Michael Elabd, Tom Erez, Xiaolin Fang, Claudio Fantacci, Cody Fong, Erik Frey, Chuyuan Fu, Ruiqi Gao, Marissa Giustina, Keerthana Gopalakrishnan, Laura Graesser, Oliver Groth, Agrim Gupta, Roland Hafner, Steven Hansen, Leonard Hasenclever, Sam Haves, Nicolas Heess, Brandon Hernaez, Alex Hofer, Jasmine Hsu, Lu Huang, Sandy H. Huang, Atil Iscen, Mithun George Jacob, Deepali Jain, Sally Jesmonth, Abhishek Jindal, Ryan Julian, Dmitry Kalashnikov, M. Emre Karagozler, Stefani Karp, Matija Kecman, J. Chase Kew, Donnie Kim, Frank Kim, Junkyung Kim, Thomas Kipf, Sean Kirmani, Ksenia Konyushkova, Li Yang Ku, Yuheng Kuang, Thomas Lampe, Antoine Laurens, Tuan Anh Le, Isabel Leal, Alex X. Lee, Tsang-Wei Edward Lee, Guy Lever, Jacky Liang, Li-Heng Lin, Fangchen Liu, Shangbang Long, Caden Lu, Sharath Maddineni, Anirudha Majumdar, Kevis-Kokitsi Maninis, Andrew Marmon, Sergio Martinez, Assaf Hurwitz Michaely, Niko Milonopoulos, Joss Moore, Robert Moreno, Michael Neunert, Francesco Nori, Joy Ortiz, Kenneth Oslund, Carolina Parada, Emilio Parisotto, Amaris Paryag, Acorn Pooley, Thomas Power, Alessio Quaglino, Haroon Qureshi, Rajkumar Vasudeva Raju, Helen Ran, Dushyant Rao, Kanishka Rao, Isaac Reid, David Rendleman, Krista Reymann, Miguel Rivas, Francesco Romano, Yulia Rubanova, Peter Pastor Sampedro, Pannag R Sanketi, Dhruv Shah, Mohit Sharma, Kathryn Shea, Mohit Shridhar, Charles Shu, Vikas Sindhwani, Sumeet Singh, Radu Soricut, Rachel Sterneck, Ian Storz, Razvan Surdulescu, Jie Tan, Jonathan Tompson, Saran Tunyasuvunakool, Jake Varley, Grace Vesom, Giulia Vezzani, Maria Bauza Villalonga, Oriol Vinyals, René Wagner, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Chengda Wu, Markus Wulfmeier, Fei Xia, Ted Xiao, Annie Xie, Jinyu Xie, Peng Xu, Sichun Xu, Ying Xu, Zhuo Xu, Jimmy Yan, Sherry Yang, Skye Yang, Yuxiang Yang, Hiu Hong Yu, Wenhao Yu, Wentao Yuan, Yuan Yuan, Jingwei Zhang, Tingnan Zhang, Zhiyuan Zhang, Allan Zhou, Guangyao Zhou, and Yuxiang Zhou. Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer, 2025. URL https://arxiv.org/abs/2510.03342.

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.

Meng-Hao Guo, Zhe-Han Mo, Jia-Jun Wang, Yi Zhang, Kejin Wang, Yi-Xuan Deng, Jia-Peng Zhang, Yongming Rao, and Shi-Min Hu. Transferring the intelligence of VLMs to robotic control, 2026a. URL https://arxiv.org/abs/2609.22966.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=748bHL2BAv.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loïc Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. DreamGen: Unlocking generalization in robot learning through video world models. In Joseph Lim, Shuran Song, and Hae-Won Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 5170–5194. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/jang25a.html.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7482–7491, Los Alamitos, CA, USA, June 2018. IEEE Computer Society. doi: 10.1109/CVPR.2018.00781. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00781.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. 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.

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 Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=wPEIStHxYH.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/lee19d.html.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal World Modeling for Robot Control. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.016.

Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 3943–3960. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/liang25b.html.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 44776–44791. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8c3c666820ea055a77726d66fc7d447f-Paper-Datasets_and_Benchmarks.pdf.

Xin Liu, Haoran Li, and Dongbin Zhao. Videos are sample-efficient supervisions: Behavior cloning from videos via latent representations. In D. Belgrave, C. Zhang, H. Lin, L. Montoya, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, pages 157162–157193. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/e67a667df97d44fa43ffe8d179b44dd8-Paper-Conference.pdf.

Zeyi Liu, Shuang Li, Eric Cousineau, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Geometry-aware 4D video generation for robot manipulation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=18gC6pZVVc.

Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, Cui Wenbo, Senmao Qi, Shuo Wang, Yixin Zheng, Mi Yan, Xuesong Shi, Haoran Li, Dongbin Zhao, Ming-Yu Liu, Zhizheng Zhang, Li Yi, Yizhou Wang, and He Wang. LDA-1B: Scaling Latent Dynamics Action Model via Universal Embodied Data Ingestion. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.210.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction, 2020. URL https://arxiv.org/abs/1802.03426.

NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical AI, 2025a. URL https://arxiv.org/abs/2501.03575.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots, 2025b. URL https://arxiv.org/abs/2503.14734.

OpenAI. GPT-5.5 system card, 2026. URL https://openai.com/index/gpt-5-5-system-card/.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi Jim Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi,

Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick Tree Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

Shivansh Patel, Shraddhaa Mohan, Hanlin Mai, Unnat Jain, Svetlana Lazebnik, and Yunzhu Li. Robotic manipulation by imitating generated videos without physical demonstrations. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=tv0Sz8A9Tc.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities, 2026. URL https://arxiv.org/abs/2604.15483.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=rvUq3cxpDF.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michael Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Herve Jegou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=2NlGyqNjns. Featured Certification.

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=meRCKuUpmc.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. URL https://arxiv.org/abs/2503.20314.

Yifan Xu, Huapeng Wei, Minxuan Lin, Yingying Deng, Kekai Sheng, Mengdan Zhang, Fan Tang, Weiming Dong, Feiyue Huang, and Changsheng Xu. Transformers in computational visual media: A survey. *Computational Visual Media*, 8(1):33–62, 2022. doi: 10.1007/s41095-021-0247-3.

Zhen Xu, Hongyu Zhou, Sida Peng, Haotong Lin, Haoyu Guo, Jiahao Shao, Peishan Yang, Qinglin Yang, Sheng Miao, Xingyi He, Yifan Wang, Yue Wang, Ruizhen Hu, Yiyi Liao, Xiaowei Zhou, and Hujun Bao. Towards depth foundation models: Recent trends in vision-based depth estimation. *Computational Visual Media*, 12 (2):243–271, 2026. doi: 10.26599/CVM.2025.9450517.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June 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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=VYOe2eBQeh.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies, 2026. URL https://arxiv.org/abs/2602.15922.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/3fe78a8acf5fda99de95303940a2420c-Paper.pdf.

Tianhe Yu, Ted Xiao, Jonathan Tompson, Austin Stone, Su Wang, Anthony Brohan, Jaspiar Singh, Clayton Tan, Dee M, Jodilyn Peralta, Karol Hausman, Brian Ichter, and Fei Xia. Scaling Robot Learning with Semantically Imagined Experience. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.027.

Jun Yuan, Changjian Chen, Weikai Yang, Mengchen Liu, Jiazhi Xia, and Shixia Liu. A survey of visual analytics techniques for machine learning. *Computational Visual Media*, 7(1):3–36, 2021. doi: 10.1007/s41095-020-0191-7.

Hongtao Zheng and Xinyan Huang. Photorealistic fire scene video generation via multimodal large language model and pre-trained video diffusion model. *Computational Visual Media*, 12(3):841–848, 2026. doi: 10.26599/CVM.2025.9450511.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Tai Wang, Ya-Qin Zhang, Jingjing Liu, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kt51kZH4aG.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified World Models: Coupling Video and Action Diffusion for Pretraining on Large Robotic Datasets. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025a. doi: 10.15607/RSS.2025.XXI.015.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning, 2025b. URL https://arxiv.org/abs/2009.12293.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Hyperparameters and Training Details

Table 2, Table 3, and Table 4 list the full hyperparameter configurations for the three training stages. All experiments are run on a single node with 8 NVIDIA RTX 4090 GPUs (24 GB each). End-to-end wall-clock training time is approximately 19 hours for stage 1, 120 hours for stage 2, and 7 hours for stage 3.

Table 2: Stage 1 (LAM Training) hyperparameters.

Component	**Setting**
Visual backbone	DINOv3 ViT-B/16 (frozen)
Encoder / decoder	12 layers, dim 768, 12 heads
VQ codebook size C	64
Latent dimension	128
Tokens per frame pair K	4
Action Decoder	3-layer MLP, hidden $[256, 128]$, GELU
Training data	DROID
Global batch size	512 (64 per GPU)
Optimizer	AdamW (lr 10^{-4}, wd 10^{-2})
Gradient clipping	0.1
Epochs	1
Action loss weight λ_{act}	0.01

Table 3: Stage 2 (VLA Pretraining) hyperparameters.

Component	**Setting**
Backbone	Qwen3-VL 4B-Instruct
Visual encoder	frozen
Projector	full precision, fully trained
Language model	LoRA (rank 64, $\alpha = 128$)
Global batch size	256
Optimizer	AdamW (lr 5×10^{-5}, wd 10^{-2})
Schedule	constant with 3000-step warmup
Parallelism	DeepSpeed ZeRO-2
Epochs	2
Latent tokenization	on-the-fly (lazy encoding)

Table 4: Stage 3 (Deployment Adaptation) hyperparameters.

Component	**Setting**
Initialization	Stage 2 checkpoint
Action head hidden dim	512
Backbone learning rate	10^{-4} (cosine decay to 10% of peak)
Action head learning rate	3.5×10^{-4} (wd 10^{-3})
Global batch size	256
Epochs	75
Warmup steps	200
Generated data weight λ_v	0.1
Action window W	16
Gradient projection	enabled at every step

B Data and Task Details

DROID (Stages 1 and 2) The DROID dataset provides approximately 362 hours of teleoperated manipulation across 564 scenes. The $\sim$40% of trajectories with precise camera extrinsics form the calibrated subset on which the action loss $\mathcal{L}_{\text{act}}$ is applied; the remaining trajectories contribute only to the reconstruction and VQ objectives.

LIBERO (Stage 3 source) Source data $\mathcal{D}_{\text{src}}$ is drawn from LIBERO [Liu et al., 2023]. We use its standard task suite as the source-domain action-labeled data for joint finetuning.

Target environment (Stage 3) The target environment is a custom robosuite [Zhu et al., 2025b] scene that differs from the source in lighting, desk textures, object identities and layout, and task instruction. Target-domain videos $\mathcal{D}_{\text{gen}}$ are synthesized with Wan 2.6 [Wan et al., 2025], conditioned on the initial frame and the language instruction, yielding 20 rollouts per task.

Target task suite The five target tasks instantiate a single unseen manipulation skill (enclosing power grasp followed by short-range relocation) under systematically varied spatial contexts. This controlled design is deliberate: by fixing the grasp type and success criterion, we isolate how well generated-video supervision generalizes across different approach geometries without confounding from semantic task variation. The five variants differ in support surface, surrounding distractors, and spatial relation between the target object and nearby landmarks, producing meaningfully different trajectory requirements (collision avoidance, depth estimation, pre-grasp alignment). Table 5 lists the instruction templates.

Table 5: Target task suite. Each task instantiates the same unseen-object relocation skill under a different spatial configuration, varying the approach geometry, support relation, and surrounding distractors while keeping the target object and success criterion controlled. `[obj]`, `[ref]`, and `[goal]` denote the target object, reference landmark(s), and goal location, respectively.

Task	Spatial context	Instruction template
A	Between distractors	Pick up `[obj]` between $\texttt{[ref}_1\texttt{]}$ and $\texttt{[ref}_2\texttt{]}$ and place it on `[goal]`.
B	Open table surface	Pick up `[obj]` from `[ref]` and place it on `[goal]`.
C	On supporting object	Pick up `[obj]` on `[ref]` and place it on `[goal]`.
D	Adjacent to distractor	Pick up `[obj]` next to `[ref]` and place it on `[goal]`.
E	Depth-ambiguous region	Pick up `[obj]` on `[ref]` and place it on `[goal]`.

C MLLM Evaluation Prompt

In RQ3 (Sec. 4.4), we use GPT-5.5 as a rollout evaluator. The model receives the full rollout video along with the task instruction and returns a composite score (five dimensions, each 0–4, totaling 20 points) assessing task completion and execution quality. The prompt template is shown below.

```
You will watch a rollout video of a robot completing the <task objective> and
    score it based on the quality of the task execution process.
Please inspect the video frame by frame. Do not score based only on key frames,
    sampled frames, or an overall impression. Before scoring, you must clearly
    observe the continuous process from the robot approaching the target object,
    making contact, performing the operation, and reaching the end of the task.
    Pay close attention to the approach path, pauses, jitter, incorrect contact,
    corrective actions, environmental collisions, operation effect, and final
    state.
Please score the performance across the following five dimensions. Each dimension
    is scored from 0 to 4, for a total of 20 points. Each score must be an
    integer and include a brief reason.

(1) Approach and Initial Operation Smoothness
Evaluate whether the robot's process from approaching the target object to
    forming effective contact is natural, continuous, and accurate.
```

```
4 points: The robot approaches accurately in one attempt and forms effective
    contact. The path is natural and continuous, with no obvious jitter, pauses,
    probing, incorrect contact, or secondary adjustment.
3 points: Overall fairly smooth, with only slight deceleration, small pose
    correction, or very brief alignment. No obvious pause, hesitation, jitter,
    poking, pushing, retreating, or post-contact readjustment.
2 points: Average process, with visible pauses, hesitation, jitter, path
    deviation, obvious pose correction, brief incorrect stopping, slight poking,
    pushing, or post-contact adjustment. Effective contact is eventually formed.
1 point: Effective contact is formed only after multiple pokes, pushes, probing
    actions, retreats, or repeated adjustments.
0 points: The robot is unable to form effective contact for a long time,
    repeatedly collides with or pushes away the target object, or makes
    incorrect contact; the initial stage is chaotic.

(2) Contact Position Accuracy and Operation Quality
Evaluate whether the robot's contact position, pose, contact method, force
    direction, and stability are reasonable.
4 points: The contact position is ideal, the pose and force application are
    reasonable, contact is stable and sufficient, and the target object shows no
    obvious sliding, rotation, tilting, or instability.
3 points: The contact position is generally accurate, with only slight offset or
    slightly insufficient contact. The target object remains basically stable,
    with no obvious sliding, rotation, tilting, slipping, or instability risk.
2 points: The contact position is not ideal, with obvious offset, insufficient
    contact area, poor force direction, or uneven force, causing visible
    tilting, sliding, rotation, slight slipping, or instability of the target
    object. The task can still continue.
1 point: The contact position is poor, contacting only an edge, a small area, or
    the wrong area. The target object is clearly unstable, nearly drops, gets
    stuck, or fails.
0 points: The contact position is extremely poor, causing clear operation
    failure, severe instability, or requiring recovery from a failed state.

(3) Execution Stability and Trajectory Smoothness
Evaluate whether the main execution phase is stable, smooth, and continuous. Do
    not separately evaluate environmental collisions here.
4 points: The execution process is continuous and natural, with steady speed and
    no obvious jitter, sudden stops, retreats, detours, or unnecessary
    adjustments. The target object and end effector remain stable.
3 points: Overall stable, with only slight speed changes, small trajectory
    adjustments, very brief pauses, or slight shaking. No obvious pose changes,
    instability, or unnecessary corrections.
2 points: The process is not very smooth, with obvious pauses, jitter, uneven
    speed, path corrections, pose adjustments, repeated micro-adjustments, brief
    retreats, or unnecessary detours. The target object or end effector visibly
    shakes, tilts, or becomes unstable, but the task can still continue.
1 point: Multiple jitters, sudden stops, retreats, or repeated adjustments occur.
    The trajectory is clearly unnatural, and the target object or end effector
    is obviously unstable.
0 points: The execution process is chaotic, with long stagnation, severe jitter,
    or multiple ineffective movements. The task state is severely unstable or
    close to losing control.

(4) Environmental Interaction and Collision Control
Evaluate whether the robot, end effector, or target object makes unnecessary
    contact with the environment and causes disturbances.
4 points: No obvious collision or unnecessary contact throughout the process; no
    non-target objects or environmental structures are disturbed.
3 points: Only slight, brief scraping or near-contact occurs, without visible
    movement, shaking, squeezing, dragging, mistaken contact, or task
    interference.
2 points: Obvious but not severe collision, scraping, pressing, dragging, or
    mistaken contact occurs, causing small movement, shaking, pose change, brief
    obstruction, or slight interference, but does not lead to failure.
```

```
1 point: The robot collides with environmental objects multiple times, or clearly
    pushes, squeezes, drags, or mistakenly contacts objects, causing noticeable
    disturbance or affecting the task.
0 points: Severe collision or major environmental disturbance occurs, causing
    objects to be knocked away, dragged, toppled, mistakenly triggered, or
    seriously interfering with the task.

(5) Task Success and Final State Accuracy
Evaluate whether the task is completed and whether the final result reaches the
    target position or target state.
4 points: The task succeeds. The final result is accurately in place and stable,
    with no obvious deviation, over-operation, or risk of failure.
3 points: The task succeeds. The final result is basically in place, with only
    slight position or pose deviation. The target object is stable, with no
    obvious edge offset, tilting, slipping, or subsequent instability risk.
2 points: The task is basically successful, but the final position, pose, or
    state is clearly not ideal, such as obvious deviation from the target
    center, edge offset, tilting, incomplete placement, or risk of slipping,
    toppling, deviation, rebound, or later instability.
1 point: The task barely succeeds or is close to succeeding. The final state is
    unstable, clearly deviates from the target, or only part of the objective is
    completed.
0 points: The task fails. The target position or state is not reached, or
    dropping, non-triggering, severe deviation, jamming, instability, or another
    issue prevents task completion.

Please output in the following format:
(1) Approach and Initial Operation Smoothness: X/4
Reason: ...
(2) Contact Position Accuracy and Operation Quality: X/4
Reason: ...
(3) Execution Stability and Trajectory Smoothness: X/4
Reason: ...
(4) Environmental Interaction and Collision Control: X/4
Reason: ...
(5) Task Success and Final State Accuracy: X/4
Reason: ...
Total Score: X/20
```

D Additional Results

Table 6 gives the per-task results of the tokenizer-swap experiment of Sec. 4.3.

Table 6: **Tokenizer swap.** Success rate (%) when the official LAPA and UniVLA latent action models replace ours in an otherwise identical stage-2/3 pipeline on the same training data. Each entry averages 50 closed-loop rollouts.

	Tokenizer	A	B	C	D	E	Avg
Source Env	UniVLA	74	96	100	94	80	88.8
	LAPA	78	84	92	92	90	87.2
	Ours	96	98	98	100	88	**96.0**
Target Env (adapted)	UniVLA	14	4	0	0	2	4.0
	LAPA	10	32	18	0	0	12.0
	Ours	36	78	80	82	8	**56.8**

Figures 8–11 provide per-axis and magnitude UMAP visualizations supplementing the codebook analysis in Sec. 4.2. Our codebook exhibits smooth, axis-aligned gradients across all translational dimensions, whereas LAPA and UniVLA show fragmented clusters with no consistent directional structure.

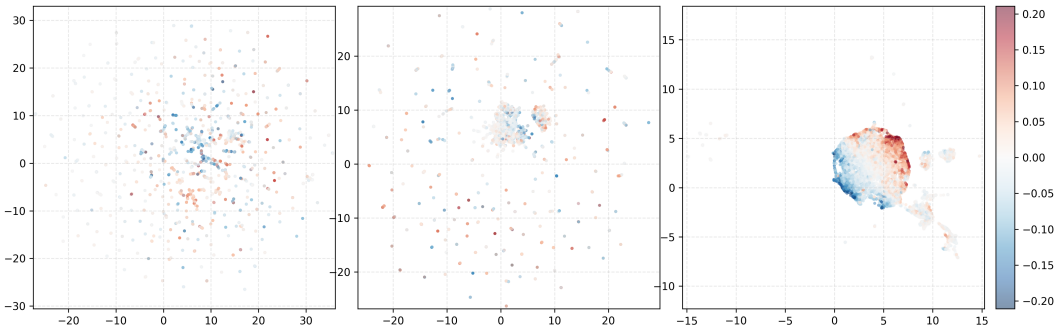


Figure 8: UMAP visualization colored by X-axis displacement. **Left**: LAPA; **Middle**: UniVLA; **Right**: Ours.

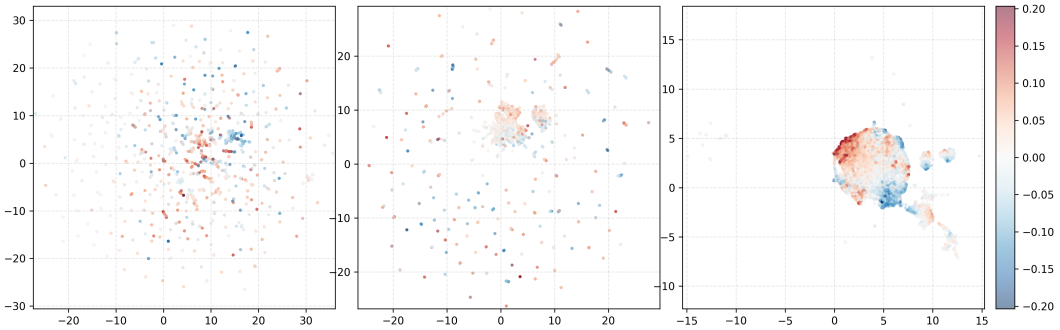


Figure 9: UMAP visualization colored by Y-axis displacement. **Left**: LAPA; **Middle**: UniVLA; **Right**: Ours.

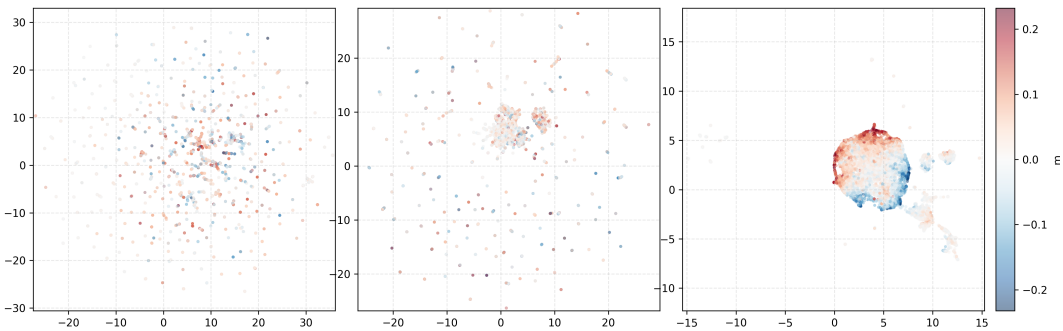


Figure 10: UMAP visualization colored by Z-axis displacement. **Left**: LAPA; **Middle**: UniVLA; **Right**: Ours.

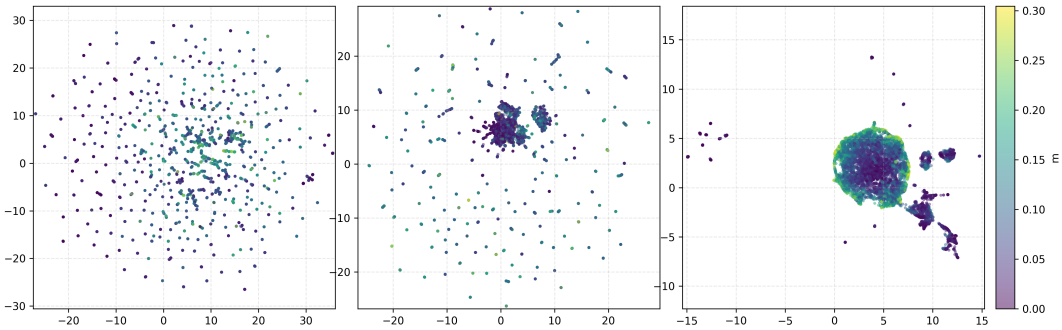


Figure 11: UMAP visualization colored by displacement magnitude. **Left**: LAPA; **Middle**: UniVLA; **Right**: Ours.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the paper's scope and main contributions: the three-stage FreeAct framework, physically grounded latent action learning, demonstration-free adaptation from generated videos, and the self-evolution extension. The stated claims are supported by the method and experimental results, and are scoped to the evaluated settings.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in the *Limitations and future work* paragraph, including the evaluation scope and directions for improving generation fidelity.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present theoretical results, theorems, or formal proofs. The equations in the method section define model objectives, losses, metrics, and optimization updates, and their assumptions and usage are described in the corresponding method and experiment sections.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper describes the datasets, baselines, evaluation protocol, model components, losses, training stages, and main hyperparameters needed to reproduce the experimental results. Additional architectural details, training schedules, and implementation details are provided in the appendix. Code files will be released to further support reproducibility.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release the code and reproduction instructions, including environment setup, data preparation, training and evaluation commands, and scripts for reproducing the experimental results. The paper uses public datasets where applicable and describes how generated videos and intermediate pseudo-labels are produced.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the datasets, data splits, baselines, evaluation protocol, model architecture, training stages, losses, and key hyperparameters in the main text, with full training schedules, optimizer settings, and implementation details provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Each policy success rate is averaged over 50 independent closed-loop rollouts, giving a binomial standard error of at most 7.1 percentage points. The main improvements, such as 2% to 57% and 57% to 74%, are much larger than this uncertainty. For Table 1, held-out test results are reported to verify generalization beyond the training split.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the hardware used for all experiments, including 8 NVIDIA RTX 4090 GPUs with 24GB memory each, and provides the wall-clock training time for each training stage.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work conforms to the NeurIPS Code of Ethics and does not involve human subjects or sensitive personal data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The work lowers the data-collection barrier for deploying manipulation policies in new environments. Potential risks of imperfect generated supervision are discussed alongside our experimental analysis.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models, scraped datasets, or generative models requiring additional safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper credits the existing datasets, models, and codebases used in the experiments, and the released code will include their licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The released code and any new assets introduced by the paper will be documented with setup instructions, training and evaluation details, licenses, and usage notes.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We use an MLLM (GPT-5.5) as an automated rollout evaluator to score policy rollouts for self-evolution (Sec. 4.4); its usage is described in the main text and the *MLLM Evaluation Prompt* appendix.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.