

World2Act: Latent Action Post-Training from World Model Dynamics

Abstract—World Models (WMs) offer a promising mechanism for post-training Vision-Language-Action (VLA) policies by providing dynamics priors that improve generalization under task and scene variation. However, most WM-based post-training methods rely on pixel-space supervision, making policies sensitive to visual artifacts introduced by imperfect WM rollouts. We present WORLD2ACT, a latent-space post-training framework that transfers WM dynamics to the VLA policy without pixel-space supervision. WORLD2ACT operates in two stages: (i) it induces a shared video-action latent space by contrastively aligning WM-dynamics latents with action embeddings, and (ii) it post-trains the VLA by guiding policy action representations toward WM-imagined dynamics rather than decoded pixels. Built on GR00T-N1.6, WORLD2ACT delivers absolute success-rate gains of up to +2.5% on simulation benchmarks (RoboCasa, LIBERO, Bridge-SIMPLER) and +5.0% on a real robot over finetuned VLA baselines. Notably, it outperforms pixel-space WM supervision by up to +6.0%, including on LIBERO where pixel supervision *degrades* the baseline, suggesting that latent WM dynamics offer a more stable WM-based post-training alternative to pixel-space transfer. Code and checkpoints will be released to support future research on WM-based post-training.

I. INTRODUCTION

World models (WMs) provide dynamics priors to improve decision-making in manipulation tasks [2]. These priors can complement the expert demonstrations used to train vision-language-action (VLA) models [3]. This motivates WM-based post-training, in which pretrained VLA policies are further optimized using synthetic experience from WMs [4], [5]. This paper studies how to translate WM predictions into reliable supervision for post-training VLA policies.

Several WM-based post-training methods derive learning signals from generated video rollouts, either by inferring pseudo-actions with an inverse dynamics model (IDM) [6] or by calculating rewards for policy optimization [4], [7]. However, because these signals are directly computed from WM rollouts, their quality is limited by the fidelity of the generated pixels. When rollouts contain visual artifacts [8], or inconsistent contacts [9], these errors can be converted into noisy action labels or rewards, compounding over long horizons [10]. Such noisy training signals can degrade the performance of finetuned VLA policies, necessitating more reliable post-training approaches.

Video latents from WMs capture transferable behavioral dynamics [11], offering a natural alternative to pixels for transferring knowledge from WMs to VLAs [12]. Prior latent-video-to-action methods typically learn unified video-action representations, such as Cosmos Policy [2] and τ_0 -WM [13]. However, coupling video prediction and action generation can be computationally expensive [14]. We instead enable efficient VLA post-training by freezing the

WM and VLA backbones and training a lightweight residual policy to match policy action latents to WM-predicted dynamics targets in a shared low-dimensional video-action latent space. This bypasses pixel-space action refinement and reduces sensitivity to visual artifacts in WM rollouts.

We introduce WORLD2ACT, a latent post-training method that aligns VLA action representation with WM latent video dynamics. Our method proceeds in two stages. First, we train video and action adapters that project WM video latents and expert action sequences into a shared representation space using a contrastive matching objective. Second, we train a lightweight residual policy to refine GR00T-N1.6’s predicted actions within the learned video-action latent space while keeping the VLA backbone frozen. Across RoboCasa [15], LIBERO [16], Bridge-SIMPLER [17], and real-world robot evaluations, WORLD2ACT improves post-training performance over pixel-space supervision, as illustrated in Fig. 1. In summary, our contributions are threefold:

- We identify sensitivity to visual artifacts in imagined rollouts as a key limitation of pixel-space WM post-training, motivating latent-space dynamics alignment as a more stable alternative.
- We introduce WORLD2ACT, a two-stage post-training method that aligns VLA action representations with WM video-dynamics latents through a shared video-action adapter space.
- Built on GR00T-N1.6, WORLD2ACT consistently improves policy performance across simulation [15]–[17] and real-world benchmarks.

II. RELATED WORK

World Models for Policy Improvement. WMs have emerged as a key component in developing generalist robots [18]. Because real-world data collection is serial, slow, and labor-intensive, WMs provide a low-cost, resettable, and highly scalable alternative for simulating environments and finetuning policies [19]. One line of work [20] uses interactive, action-conditioned WMs for model-based planning [2], [12] and policy optimization [5], [8], [21]. These approaches use WMs as virtual environments to construct imagined trajectories conditioned on candidate or policy-generated actions, a process that can be computationally expensive [22]. Another line of work [6], [23], [24] uses instruction-conditioned video WMs to generate future visual trajectories from initial observations, without conditioning generation on policy-generated actions. Following this instruction-conditioned paradigm, our paper uses WM-predicted latent dynamics to supervise a lightweight residual policy, enabling efficient VLA post-training.

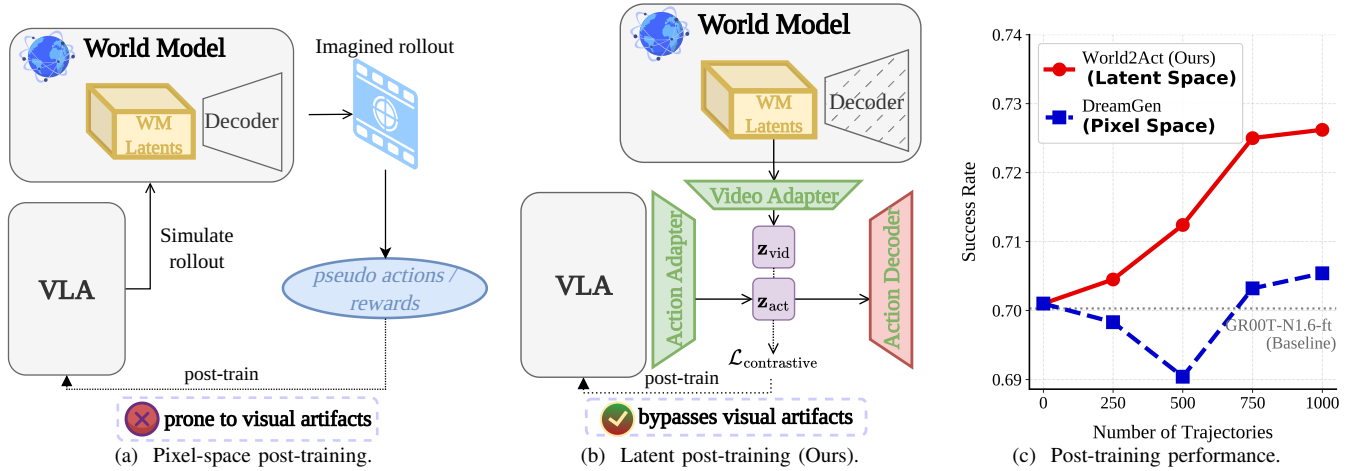


Fig. 1. (a) Problem: Prior WM $\rightarrow$ VLA post-training relies on imagined pixel rollouts, making policies sensitive to visual artifacts. (b) Solution: We bypass this by directly aligning the VLA action representation with the WM latent space. (c) Post-training on GR00T-N1.6 [1] indicates that our latent post-training yields consistent improvements, overcoming the instability of pixel-space methods.

Post-Training Vision-Language-Action Models. VLA models are commonly trained on large-scale demonstration datasets and then finetuned on task- or robot-specific data [5]. However, collecting expert demonstrations across diverse tasks and environments is costly, leaving gaps in training coverage that can limit generalization to unseen behaviors and scene configurations [6]. Recent work, therefore, explores WM-based post-training, using synthetic experience to improve pretrained VLA policies beyond supervised finetuning [5]. DreamGen [6] uses IDMs to label generated videos with pseudo-actions for policy training. Other methods compute rewards from imagined visual rollouts to guide policy improvement [4], [5], [7], [8], [19]–[21]. Ctrl-World [22] selects successful imagined trajectories for finetuning using human judgments, which requires manual evaluation of generated rollouts. WORLD2ACT uses WM-predicted video latents to guide residual policy updates through alignment in a shared video–action space.

Latent Actions. Latent action representations provide effective interfaces for learning from offline robot data and unlabeled videos [11], [25], [26]. LAPA [27] learns latent actions from human manipulation videos for VLA pre-training. CLAP [28] extends this line of work by aligning visual transitions with robot action latents via contrastive learning. Although these methods support cross-embodiment learning, human motions do not always map directly to robot actions, which can limit skill transfer [29]. Recent methods explore coupling latent actions with video prediction, using paired videos and actions from the same embodiment and environment to transfer visual-dynamics knowledge to robot control [2], [30]. Other approaches steer action learning through future-observation supervision: FLARE [31] aligns intermediate policy features with embeddings of future images, while CoWVLA [32] combines latent-motion supervision with terminal keyframe prediction. Our method uses WM-predicted video latents to guide residual post-training of an existing VLA, aligning action and video representations chunk by chunk in a shared latent space.

III. METHOD

We present WORLD2ACT, a post-training method that transfers visual-dynamic priors from a frozen instruction-conditioned video WM to a VLA policy, as shown in Fig. 2. We first describe the WM latent representation, then present our two-stage pipeline: *i) Stage 1*: learning a shared video–action latent space that aligns WM latent dynamics with robot actions via bridge adapters, and *ii) Stage 2*: training a lightweight residual policy on top of a frozen VLA, guided by the latent space learned in Stage 1.

A. Preliminary: Instruction-Conditioned Video World Model

We use Cosmos-Predict2 [33], an instruction-conditioned video WM that predicts future visual states in the latent space of a pretrained video tokenizer. During WM training, the video tokenizer encodes ground-truth videos into latent trajectories $\mathbf{V}^{\text{gt}}$, and a denoising network is trained via flow matching [33] to transport noisy samples toward the distribution of clean video latents. We freeze both the video tokenizer and the denoising network throughout policy learning to preserve the pretrained latent video-dynamics priors.

Given an initial observation s_1 and language instruction ℓ , we draw an initial noise sample ϵ and run a fixed sampling procedure to generate a denoised latent trajectory. We denote the resulting conditional sampling procedure by $\mathcal{W}$:

$$\mathbf{V} = \mathcal{W}(\epsilon; s_1, \ell), \quad \mathbf{V} = \{V_t\}_{t=1}^T. \quad (1)$$

Each V_t is a video latent chunk of the predicted rollout. Since Cosmos-Predict2 is trained on temporally ordered ground-truth video latents [33], we treat V_t as an approximate proxy for V_t^{gt} , which is temporally paired with the corresponding M -step action chunk. We use $\mathbf{V}$ directly as the visual-dynamics target without pixel decoding.

B. Stage 1: Aligning Video Dynamics and Robot Actions

Stage 1 learns a video–action latent bridge from expert demonstrations. For each demonstration, the video WM generates $\mathbf{V} = \{V_t\}_{t=1}^T$ using only (s_1, ℓ) , while the corresponding expert actions are partitioned into T chunks of M steps and paired with the WM latents by temporal index.

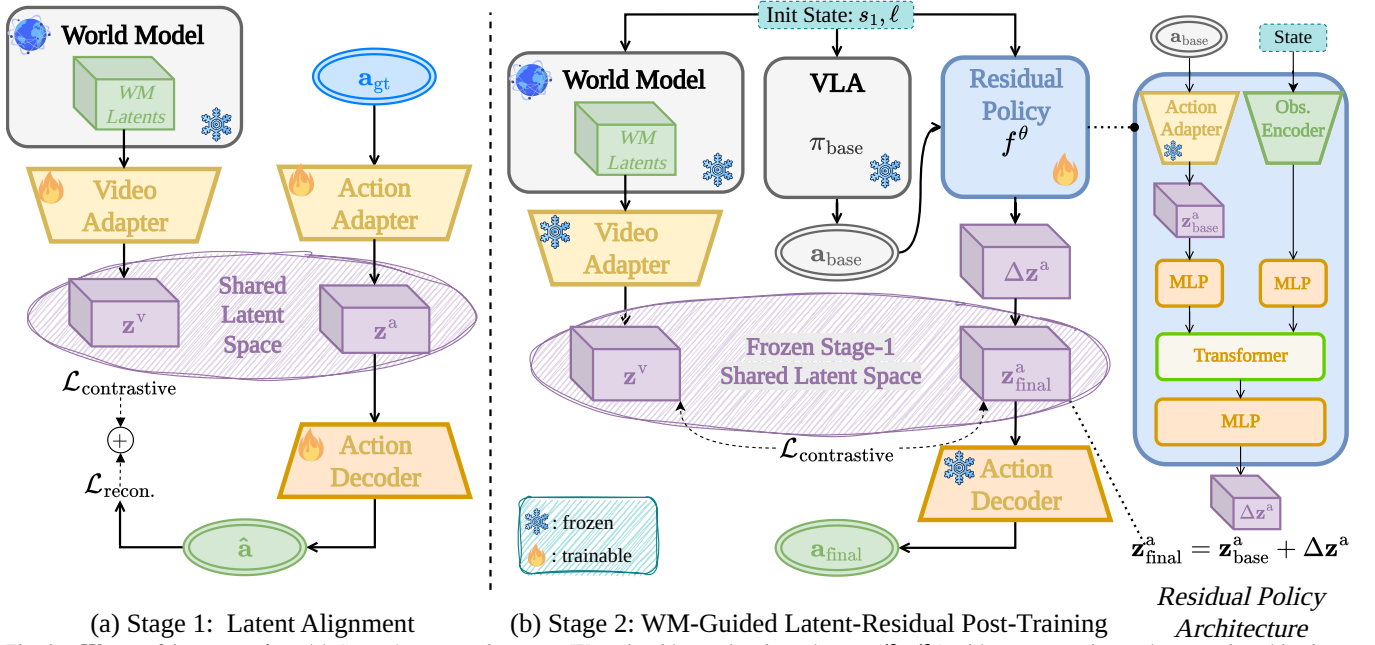


Fig. 2. **WORLD2ACT overview.** (a) *Stage 1: Latent alignment.* We train video and action adapters ($\mathcal{B}_v, \mathcal{B}_a$) with reconstruction and contrastive objectives. (b) *Stage 2: VLA post-training.* We freeze the VLA and learn a residual policy guided by WM-induced latent dynamics.

A video adapter $\mathcal{B}_v$ maps each WM latent to a D -dimensional vector, producing $\mathbf{z}^v = \{z_t^v\}_{t=1}^T \in \mathbb{R}^{T \times D}$, where $z_t^v = \mathcal{B}_v(V_t)$. In parallel, an action adapter $\mathcal{B}_a$ partitions $\mathbf{a}_{gt}$ into M -step chunks $\bar{\mathbf{a}}_t^{gt} = [a_{(t-1)M+1}^{gt}; \dots; a_{tM}^{gt}]$ and maps each chunk to $z_t^a = \mathcal{B}_a(\bar{\mathbf{a}}_t^{gt})$, yielding $\mathbf{z}^a = \{z_t^a\}_{t=1}^T \in \mathbb{R}^{T \times D}$. The resulting video and action embeddings are paired at matching chunk indices, enabling temporal alignment through corresponding chunks rather than a single global trajectory descriptor (Fig. 2a).

To encourage the action latents to preserve meaningful kinematic structure, we introduce an MLP-based *Action Decoder* $\mathcal{D}_a$, which maps each action chunk latent z_t^a back to M low-level action vectors. We train $\mathcal{B}_v$, $\mathcal{B}_a$, and $\mathcal{D}_a$ with two objectives. First, we use a reconstruction objective: unrolling the decoded chunks gives the reconstructed action sequence $\hat{\mathbf{a}}$, supervised by $\mathcal{L}_{recon} = \|\mathbf{a}_{gt} - \hat{\mathbf{a}}\|^2$.

Second, we align video and action trajectories with a bidirectional InfoNCE loss [34]. We compute trajectory similarity as the mean cosine similarity over temporally aligned chunks. For each positive pair $(\mathbf{z}_i^v, \mathbf{z}_i^a)$ from the same demonstration, trajectories from the other $B-1$ demonstrations in the minibatch serve as negatives. The loss is defined as:

$$\mathcal{L}_{contrastive} = - \sum_i \log \frac{\exp(\text{sim}(\mathbf{z}_i^v, \mathbf{z}_i^a)/\tau)}{\sum_{j=1}^B \exp(\text{sim}(\mathbf{z}_i^v, \mathbf{z}_j^a)/\tau)} - \sum_i \log \frac{\exp(\text{sim}(\mathbf{z}_i^a, \mathbf{z}_i^v)/\tau)}{\sum_{j=1}^B \exp(\text{sim}(\mathbf{z}_i^a, \mathbf{z}_j^v)/\tau)}, \quad (2)$$

where $\text{sim}(\mathbf{z}_i^v, \mathbf{z}_j^a) = \frac{1}{T} \sum_{t=1}^T \cos(z_{i,t}^v, z_{j,t}^a)$ denotes the chunk-averaged cosine similarity, and τ is a temperature hyperparameter. We set $\tau = 0.1$ following [35], and optimize $\mathcal{L} = \mathcal{L}_{recon} + \mathcal{L}_{contrastive}$. After Stage 1, $\mathcal{B}_v$, $\mathcal{B}_a$, and $\mathcal{D}_a$ are frozen to preserve a shared latent space between video dynamics and decodable robot actions.

C. Stage 2: Latent Residual Action Post-Training

Inspired by residual policy learning [36], we freeze the pretrained VLA policy π_{base} and train a lightweight *residual policy* f^θ that predicts additive corrections in the learned action-latent space. Fig. 2b shows the post-training process.

For each rollout, the WM generates a latent trajectory $\mathbf{V}$ conditioned on the initial observation s_1 and instruction ℓ . The frozen video adapter $\mathcal{B}_v$ learned in Stage 1 maps this trajectory into the shared video-action latent space, producing embeddings $\mathbf{z}^v$ that serve as supervision targets for training the residual policy f^θ .

At each chunk boundary t , the base VLA predicts an action chunk $\bar{\mathbf{a}}_{base,t} = \pi_{base}(s_t, \ell)$ from the current state s_t and instruction ℓ . The frozen action adapter $\mathcal{B}_a$ maps this chunk to the base action latent $z_{base,t}^a$. The residual policy f^θ predicts a correction $\Delta z_t^a = f^\theta(s_t, z_{base,t}^a)$, which is added to the base action latent as follows:

$$z_{base,t}^a = \mathcal{B}_a(\bar{\mathbf{a}}_{base,t}), \quad z_{final,t}^a = z_{base,t}^a + \Delta z_t^a. \quad (3)$$

The decoded action chunk $\bar{\mathbf{a}}_{final,t} = \mathcal{D}_a(z_{final,t}^a)$ is executed open-loop for M control steps before the policy is queried again at the next chunk boundary $t+1$. The Stage-1 action decoder $\mathcal{D}_a$ is kept frozen during this process, so all residual-corrected latents are decoded through the same mapping learned to reconstruct demonstration action chunks. We implement f^θ 's backbone as a transformer conditioned on observation tokens and language-conditioned base action-latent tokens, as illustrated in Fig. 2c.

To construct contrastive training pairs, we collect B rollouts under the current residual-augmented policy. For each rollout i , the action-latent trajectory $\mathbf{z}_{final,i}^a$ forms a positive pair with the WM target latents $\mathbf{z}_i^v$ generated from the same initial observation and instruction. Following the contrastive formulation in Stage 1 (Eq. (2)), we form in-batch negatives

by pairing video and action trajectories from different rollout instances, yielding $B-1$ negatives per anchor.

Finally, we perform T optimizer updates to minimize $\mathcal{L}_{\text{contrastive}}$ between the WM target embeddings and the refined action-latent trajectories across the B completed rollouts. This optimization requires neither environment rewards nor success labels. We update only the parameters of f^θ , keeping the VLA, WM and Stage-1 modules frozen.

D. Implementation Details

We use $T=24$ chunks per rollout, $M=4$ control steps per chunk, batch size $B=16$, and latent dimension $D=32$. The action dimension is $A=12$ for RoboCasa and $A=7$ otherwise, with $D_{\text{proprio}}=53$. RGB observations have resolution 256×256 , with WM latent chunks $V_t \in \mathbb{R}^{16 \times 60 \times 104}$. The bridge modules $\mathcal{B}_a$, $\mathcal{B}_v$, and $\mathcal{D}_a$ are trained for $30K$ optimizer steps with a learning rate of 10^{-5} in Stage 1 and frozen in Stage 2.

For efficient simulation post-training, we first precompute Cosmos-Predict2 target latents from initial observations and instructions, recording the corresponding scene-initialization seeds. During subsequent VLA post-training, we reuse these seeds and instructions to restore the same initial conditions for online policy rollouts. This preserves the pairing between policy trajectories and their WM targets while allowing WM generation and policy optimization to run separately.

The residual policy uses $N_{\text{src}}=3$ tokens of dimension D : *i*) the base action latent, *ii*) visual features from a CNN, and *iii*) proprioceptive features from an MLP. A transformer fuses these tokens, and an MLP head maps the contextualized base-action token to the latent correction Δz_t^a . The observation encoders, transformer, and output head together constitute f^θ . Table I summarizes the overall architecture.

TABLE I

POLICY ARCHITECTURE

Module	Architecture
Action adapter $\mathcal{B}_a$	$\text{FC}(AM, 128) - \text{GELU} - \text{FC}(128, 64) - \text{GELU} - \text{FC}(64, D)$
Action decoder $\mathcal{D}_a$	$\text{FC}(D, 64) - \text{GELU} - \text{FC}(64, 128) - \text{GELU} - \text{FC}(128, AM)$
Video adapter $\mathcal{B}_v$	$\text{Conv2D}(16, 64, k=3, s=2, p=1) - \text{GELU}$ $\text{Conv2D}(64, 128, k=3, s=2, p=1) - \text{GELU}$ $\text{GlobalAveragePooling} - \text{FC}(128, D)$
State encoder	$\text{FC}(D_{\text{proprio}}, 128) - \text{ReLU} - \text{FC}(128, D) - \text{LayerNorm}$
Visual encoder	$\text{Conv2D}(3, 16, k=8, s=4, p=0) - \text{ReLU}$; $\text{Conv2D}(16, 32, k=4, s=2, p=0) - \text{ReLU}$; $\text{Conv2D}(32, 32, k=3, s=2, p=0)$; flatten; $\text{FC}(6272, D)$
Transformer backbone	$\text{Transformer}(N_{\text{src}}=3, L=2, H=4, \text{dimension}=D)$

FC: fully connected layer; Conv2D: 2D convolution; k : kernel size; s : stride; p : padding; L : number of transformer layers; H : number of attention heads.

IV. EXPERIMENTS

Our empirical evaluation addresses three core questions:

- **Q1:** To what extent does WORLD2ACT improve VLA policy performance in simulation and on a real robot?
- **Q2:** How does latent WM supervision compare with pixel-space post-training?
- **Q3:** How do latent alignment and residual updates affect action fidelity and task success?

A. Experimental Setup

We evaluate on three simulated benchmarks: RoboCasa [15], LIBERO [16], and Bridge-SIMPLER [17], an

extension of SIMPLER [37]. Mean success rate (SR) is the primary metric. Following standard protocols [2], [15], [38], we run 50 trials per task for each of five seeds and report the mean SR.

Baselines. We benchmark against recent VLA policies: π_0 [39], $\pi_{0.5}$ [40], UVA [41], Rethink-VLA [42], OpenVLA [43], OpenVLA-OFT [44], UWM [30], UniVLA [25], Cosmos Policy [2], CoWVLA [32], FLARE [31], and GR00T-N1.6 [1]. Because GR00T-N1.6 is pretrained on mixed cross-embodiment data [45], its behavior may not match the simulator task distributions. We therefore further finetune GR00T-N1.6 on 1000 expert trajectories per simulator to obtain a task-aligned supervised reference, denoted as GR00T-N1.6-ft. Our experiments are conducted using an NVIDIA RTX 6000 Ada Generation GPU (48GB).

Post-Training Comparison. To isolate the post-training effect, we also compare WORLD2ACT with representative post-training methods: a BC variant that replaces the contrastive objective with supervised behavior cloning on the target demonstrations [46], DreamGen [6], VLA-RFT [4], and Ctrl-World [22]. All post-training methods are initialized from the same GR00T-N1.6-ft checkpoint for each simulator.

Post-Training Data Synthesis. We use Cosmos-Predict2 [33] to generate post-training trajectories, with an action-conditioned variant for VLA-RFT [4] and Ctrl-World [22]. For each simulator, we use 1000 imagined trajectories and train all post-training methods for $25K$ optimizer updates. Post-training initial scenes are disjoint from evaluation scenes.

B. Main Results

TABLE II
ROBOCASA RESULTS.

Method	Success Rate
π_0 [39]	0.625
UVA [41]	0.500
Rethink-VLA [42]	0.547
UWM [30]	0.608
FLARE [31]	0.701
Cosmos Policy [2]	0.671
GR00T-N1.5 [47]	0.641
GR00T-N1.6 [1]	0.662
GR00T-N1.6-ft [1]	0.701
+ DreamGen [6]	0.705
+ WORLD2ACT (Ours)	0.726

Results on RoboCasa. Table II shows that WORLD2ACT achieves the best RoboCasa result, improving GR00T-N1.6-ft from 70.1% to 72.6% SR, a +2.5% absolute gain. From the same finetuned initialization, it also outperforms DreamGen by +2.1%, showing that our imagined-trajectory objective is more effective than prior synthetic-data augmentation. Across all baselines, WORLD2ACT exceeds vanilla GR00T-N1.6 by +6.4% and the WM-capable Cosmos Policy by +5.5%, establishing the strongest overall performance.

Results on LIBERO. Table IV reports average SR

TABLE III
RESULTS ON BRIDGE-SIMPLER

Method	Put Spoon on Towel	Stack Green on Yellow	Put Carrot on Plate	Put Eggplant in Basket	Open Drawer	Close Drawer	Put Eggplant in Sink	Average
π_0 [39]	0.292	0.000	0.168	0.624	0.676	0.480	0.124	0.338
OpenVLA [43]	0.000	0.000	0.000	0.044	0.644	0.036	0.000	0.103
OpenVLA-OFT [44]	0.344	0.308	0.292	0.728	0.340	0.744	0.320	0.439
GR00T-N1.6-ft [1]	0.736	0.044	0.624	0.372	0.916	0.984	0.360	0.577
+ DreamGen [6]	0.720	0.048	0.648	0.424	0.928	0.976	0.344	0.584
+ WORLD2ACT (Ours)	0.768	0.044	0.664	0.388	0.916	0.992	0.364	0.591

TABLE IV
RESULTS ON LIBERO.

Method	Spatial	Object	Goal	Long	Avg.
π_0 [39]	0.968	0.988	0.958	0.852	0.942
$\pi_{0.5}$ [40]	0.988	0.982	0.980	0.924	0.969
OpenVLA-OFT [44]	0.976	0.984	0.979	0.945	0.971
UniVLA [25]	0.965	0.968	0.956	0.920	0.952
CoWVLA [32]	0.972	0.978	0.946	0.928	0.956
Cosmos Policy [2]	0.981	1.000	0.982	0.976	0.985
GR00T-N1.6-ft [1]	0.977	0.984	0.975	0.943	0.970
+ DreamGen [6]	0.991	0.826	0.992	0.876	0.921
+ WORLD2ACT (Ours)	0.995	1.000	0.988	0.940	0.981

on LIBERO. WORLD2ACT achieves 98.1% SR, whereas DreamGen degrades GR00T-N1.6-ft to 92.1%. Our method therefore yields a +6.0% absolute gain over DreamGen on the same backbone. It also surpasses $\pi_{0.5}$, OpenVLA-OFT, CoWVLA, and UniVLA, and is within 0.4% of the best reported Cosmos Policy result.

Results on Bridge-SIMPLER. Table III reports average success on Bridge-SIMPLER. We adopt AutoEval’s Bridge-SIMPLER extension [17] rather than conventional SIMPLER because its three additional tasks provide broader coverage of WidowX manipulation. Consistent with RoboCasa and LIBERO, WORLD2ACT improves GR00T-N1.6-ft from 57.7% to 59.1% SR, a +1.4% absolute gain, and outperforms DreamGen (58.4%) for the best overall result.

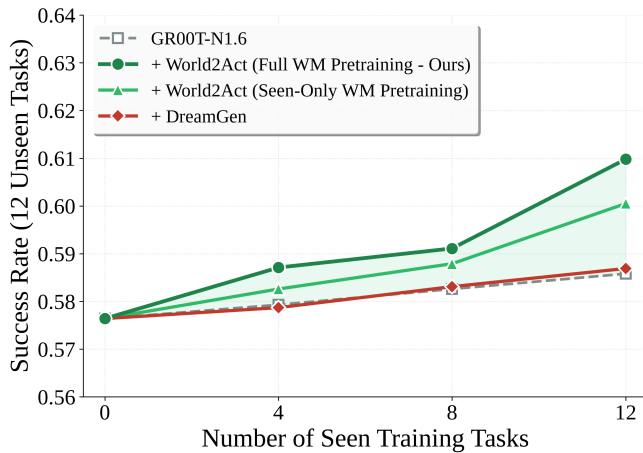


Fig. 3. Cross-task generalization.

Cross-Task Generalization. We split 24 RoboCasa tasks into 12 seen and 12 unseen tasks. The seen split covers coffee preparation, door and drawer manipulation, and microwave and stove operation. The unseen split comprises pick-and-place transfers, faucet and spout manipulation, and held-out

stove-control tasks. All methods are evaluated exclusively on the unseen split.

We continue supervised finetuning of GR00T-N1.6 as the number of seen training tasks increases from 0 to 12. Each resulting checkpoint serves as both the baseline and the shared starting point for all post-training methods. We compare DreamGen with two WORLD2ACT variants: one using a WM pretrained on all 24 tasks and another using a WM pretrained only on seen-task data.

Even with WM pretraining restricted to seen tasks, WORLD2ACT outperforms DreamGen by +1.36% (Fig. 3). Its unseen-task SR also increases with the number of seen training tasks, suggesting that WM-guided post-training transfers useful dynamics priors to the VLA across tasks.

C. Real Robot Experiments

Setup. We evaluate WORLD2ACT on a Franka Research 3 across five real-world tasks: *Pick cup and place on plate*, *Pick up bowl*, *Close microwave*, *Turn on stove*, and *Push cube*. Starting from GR00T-N1.6-ft, we use 20 real demonstrations per task collected with GELLO [48], followed by 100 WM-generated latent trajectories as post-training targets. As in simulation, the current residual-augmented policy is rolled out on the physical robot: its predicted actions are executed in the environment, and the resulting observations are fed back as policy inputs. Observations come from an external camera (Fig. 6), and we run 20 trials per task.

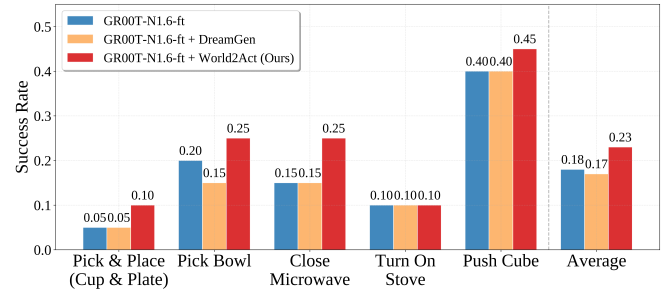


Fig. 4. Real-world success rates.

Results. Fig. 4 summarizes the real-world evaluation. WORLD2ACT achieves an average success rate of 23%, compared with 18% for GR00T-N1.6-ft and 17% for DreamGen, improving over the two baselines by +5.0% and +6.0%, respectively. It outperforms both baselines on four of the five tasks and matches their performance on turning on the stove. These results suggest that latent WM-guided post-training is more effective than synthetic-data augmentation for the real-world tasks evaluated.

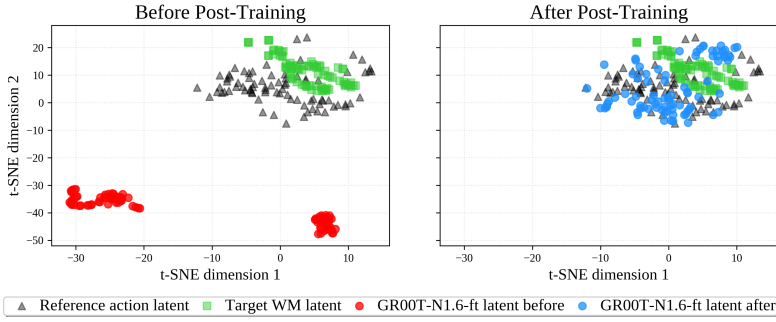


Fig. 5. Action-latent t-SNE visualization.

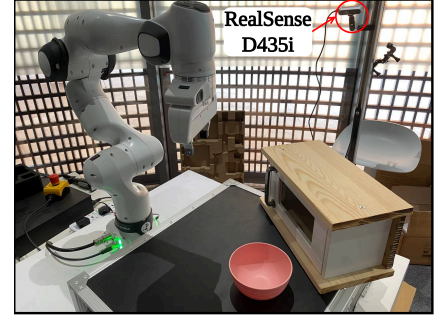
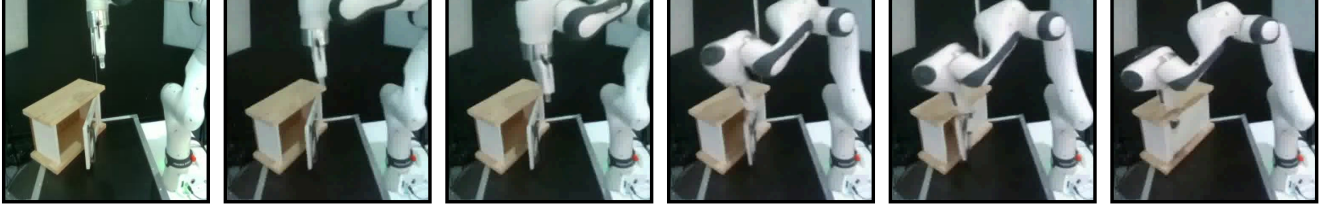


Fig. 6. Real-robot setup.

Imagination by the WM



Execution by GR00T-N1.6-ft + World2Act (Ours)

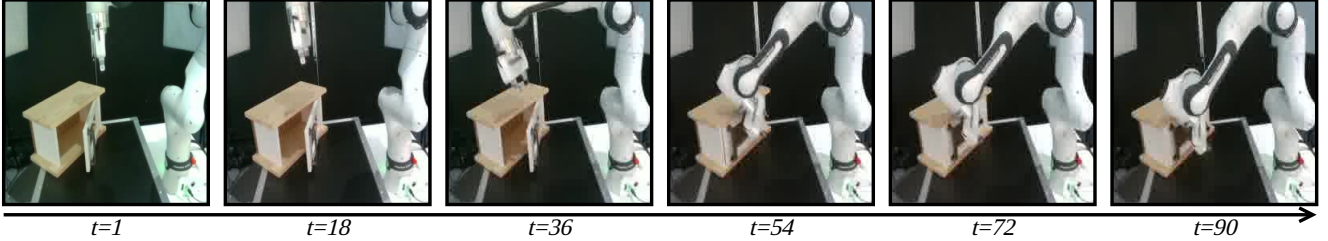


Fig. 7. WM rollout and real-robot execution.

Qualitative Evaluation. Fig. 7 shows a representative WM rollout (top) and the corresponding GR00T-N1.6-ft + WORLD2ACT execution (bottom) for *Close microwave*, where t denotes the frame timestep. Both the imagined rollout and physical execution achieve the task goal. The final imagined frame ($t=90$) contains a visual artifact in which the handle disappears. The successful physical execution illustrates that our method can use latent WM dynamics to achieve the task despite this imperfection in pixel space.

D. Latent versus Pixel Supervision

TABLE V
POST-TRAINING COMPARISON.

Method	Success Rate
GR00T-N1.6-ft [1]	0.701 ± 0.012
+ DreamGen [6]	0.705 ± 0.021
+ VLA-RFT [4]	0.710 ± 0.005
+ Ctrl-World [22]	0.698 ± 0.022
+ WORLD2ACT w/ BC [46]	0.704 ± 0.015
+ WORLD2ACT w/ GT-video latents	0.721 ± 0.010
+ WORLD2ACT (Ours)	0.726 ± 0.011

Comparison of Post-Training Methods. Table V compares post-training methods using the same GR00T-N1.6-ft backbone. Results are reported as mean $\pm$ standard deviation over five independent runs. WORLD2ACT w/ BC trains the same residual policy using target expert actions instead of WM-latent targets; DreamGen trains on IDM-labeled WM videos;

TABLE VI

RESIDUAL ACTION MSE.	
Method	MSE $\downarrow$
GR00T-N1.6-ft	0.034
+DreamGen [6]	0.030
+Zero residual ($\Delta x_t^a = 0$)	0.025
+Learned residual (Ours)	0.021

TABLE VII

FROZEN-IDM ACTION ERROR.			
IDM input	MSE $\downarrow$	MAE $\downarrow$	
GT video	0.082	0.167	
Reconstructed GT video	0.081	0.166	
Decoded WM rollout	0.097	0.199	
WM / GT	1.18 $\times$	1.19 $\times$	

VLA-RFT optimizes WM-rollout rewards; and Ctrl-World finetunes on successful action-conditioned WM rollouts. We additionally evaluate a WORLD2ACT variant that replaces generated WM targets with latents obtained by encoding temporally aligned ground-truth videos. This variant achieves 72.1% SR, close to the 72.6% obtained with generated pre-decoding WM latents, suggesting that the latent source does not substantially affect downstream performance in our setting. WORLD2ACT outperforms residual BC (70.4%), DreamGen (70.5%), VLA-RFT (71.0%), and Ctrl-World (69.8%). Furthermore, increasing the number of post-training trajectories from 0 to 1000 raises WORLD2ACT’s SR from 70.1% to 72.6%, while DreamGen exhibits non-monotonic performance across the evaluated trajectory counts (Fig. 1c). **IDM Action Error Analysis.** To investigate DreamGen’s degradation on LIBERO, we evaluate a frozen IDM on 50 held-out LIBERO-Object scenes. For each scene, we compute MSE and MAE against temporally aligned ground-truth actions over $TM = 96$ control steps (Table VII). Encoding and reconstructing ground-truth videos with the

Cosmos-Predict2 VAE leaves MSE and MAE nearly unchanged (0.081/0.166 vs. 0.082/0.167), suggesting that VAE reconstruction preserves the information used by the IDM. In contrast, WM-generated videos increase MSE and MAE by 18% and 19%, respectively, relative to ground-truth videos. These results suggest that pseudo-actions inferred from generated rollouts may introduce noisier supervision into policy post-training.

TABLE VIII

COMPONENT ABLATIONS AND POLICY-UPDATE COMPARISON.

(a) Component ablations

Study	Configuration	SR
Full model	Bidirectional InfoNCE, chunk sim., learned residual	0.726±0.011
Residual	No residual correction ($\Delta\mathbf{z}^a = 0$)	0.698±0.013
Alignment loss	Unidirectional InfoNCE [49]+chunk sim.	0.720±0.014
	Marginal contrastive loss [50]+chunk sim.	0.707±0.019
	Bidirectional InfoNCE [34]+global sim.	0.693±0.014
Target correspondence	Same task, different initial condition	0.709±0.015
	Randomly shuffled latent chunks	0.507±0.038
Target representation	Re-encoded WM video	0.719±0.015

(b) Policy-update comparison

Update	Configuration	Time	SR
Residual policy	Full model	6.8h	0.726±0.011
LoRA	Rank $r = 16$	14.6h	0.714±0.012
	Rank $r = 32$	15.3h	0.721±0.010

Post-Training from a Shared Video Source. Standard WORLD2ACT uses WM latents directly, bypassing video decoding. As an additional analysis, we evaluate a variant that receives supervision from the same WM-generated videos used by DreamGen. DreamGen converts these videos into pseudo-action labels using an IDM. Our variant instead encodes them using the frozen Cosmos-Predict2 VAE and uses the resulting latent representations to construct alignment targets, keeping the remaining post-training procedure unchanged. On RoboCasa, this variant achieves a 71.9% mean SR (Table VIII), compared with 70.5% for DreamGen, a gain of +1.4%. These results suggest that latent supervision transfers action-relevant dynamics from generated videos more effectively than IDM-based pseudo-action supervision.

E. Component Analysis and Efficiency

Does latent alignment improve action-manifold compatibility? We assess how Stage-2 WORLD2ACT latent alignment affects agreement with held-out expert actions. Table VI reports action-space MSE against temporally matched expert actions, averaged over *OpenDrawer*, *PnPCounterToCab*, and *TurnOnStove*. Post-training reduces MSE from 0.034 to 0.021, a relative reduction of 38.2%. On *OpenDrawer*, Fig. 5 additionally shows greater overlap between post-trained predicted-action latents and the reference-action and target-WM latent regions. Together, these results demonstrate improved agreement with expert actions and support our view that WM-guided latent alignment maintains com-

patibility with expert action structure.

How does the learned latent residual contribute to policy performance? To isolate the contribution of the learned residual, we set $\Delta\mathbf{z}^a = 0$ while retaining the frozen action adapter and decoder. This control reduces action MSE from 0.034 to 0.025 but leaves success essentially unchanged relative to GR00T-N1.6-ft (69.8% vs. 70.1%). Enabling the learned residual further reduces MSE to 0.021 and increases success to 72.6% (Tables VI and VIII). These results show that the learned latent residual improves both agreement with expert actions and task success compared with the policy without residual correction.

How do the alignment objective and target correspondence affect task success? We ablate the contrastive objective and perturb WM targets to examine their contributions (Table VIII). Starting from 72.6% SR, replacing bidirectional InfoNCE with unidirectional InfoNCE [49] or a marginal contrastive loss [50] reduces SR to 72.0% and 70.7%, respectively. Replacing chunk-wise matching with global trajectory similarity reduces SR to 69.3%. Disrupting temporal correspondence by randomly shuffling latent chunks causes a larger drop to 50.7%, while using targets from the same task but a different initial condition yields 70.9%. These results highlight the importance of temporal and scene correspondence, with bidirectional InfoNCE achieving the highest mean success among the evaluated objectives.

How do residual updates compare with LoRA in success rate and training time? Table VIIIb compares residual updates with LoRA finetuning. The residual policy achieves 72.6% mean SR, compared with 72.1% for LoRA at rank $r = 32$, while reducing training time from 15.3h to 6.8h, a $2.25\times$ speedup. These results support freezing the VLA and learning a lightweight latent residual as an efficient alternative to LoRA finetuning.

Remarks. Our experiments show that WORLD2ACT improves strong VLAs across simulators (Tables II, IV, III) and on a real robot (Fig. 4), with cross-task evaluations supporting generalization to unseen tasks (Fig. 3) (Q1). The comparison with other post-training baselines suggests that latent supervision transfers task dynamics from generated videos more effectively than direct pixel-space supervision (Table V). Higher IDM error on generated videos further identifies a potential limitation of pseudo-action supervision (Table VII) (Q2). Component ablations show that the learned latent residual improves agreement with expert actions and task success, while temporal and scene correspondence contribute to effective alignment (Tables VI, VIII). Compared with LoRA, residual updates improve mean success while reducing training time (Table VIIIb), supporting efficient policy refinement in the aligned latent space (Q3).

V. DISCUSSION

We introduced WORLD2ACT, a latent post-training framework that transfers generalizable dynamics priors from WMs to a GR00T-N1.6 policy, avoiding pixel-level supervision. Experiments across benchmarks show consistent performance improvements over pixel-based post-training

and other strong baselines, highlighting the promise of using WM latents as reusable dynamics priors.

Limitations. WORLD2ACT has limitations. First, contact-rich, non-prehensile tasks [51] exhibit complex contact-driven dynamics that remain difficult for WMs to capture: small errors in contact motion can compound in WM roll-outs, making the resulting priors less transferable. Second, real-world success remains modest, suggesting a persistent domain gap where complex physical dynamics are insufficiently captured by current WMs and VLAs [52].

REFERENCES

- [1] GEAR Team. (2025, Dec.) GR00T N1.6: An improved open foundation model for generalist humanoid robots. NVIDIA Research. [Online]. Available: <https://research.nvidia.com/labs/gear/gr00t-n1.6/>
- [2] M. J. Kim *et al.*, “Cosmos policy: Fine-tuning video models for visuomotor control and planning,” in *ICLR*, 2026.
- [3] L. Cao *et al.*, “Z-1: Efficient reinforcement learning for vision-language-action models,” *arXiv preprint arXiv:2606.31846*, 2026.
- [4] H. Li *et al.*, “Vla-rft: Vision-language-action reinforcement fine-tuning with verified rewards in world simulators,” *arXiv preprint arXiv:2510.00406*, 2025.
- [5] Y. Guo, T. Lee, L. X. Shi, J. Chen, P. Liang, and C. Finn, “Vlaw: Iterative co-improvement of vision-language-action policy and world model,” *arXiv preprint arXiv:2602.12063*, 2026.
- [6] J. Jang *et al.*, “Dreamgen: Unlocking generalization in robot learning through video world models,” in *CoRL*, 2025.
- [7] J. Xiao *et al.*, “World-env: Leveraging world model as a virtual environment for vla post-training,” *arXiv preprint arXiv:2509.24948*, 2025.
- [8] A. K. Sharma, Y. Sun, N. Lu, Y. Zhang, J. Liu, and S. Yang, “World-gymnast: Training robots with reinforcement learning in a world model,” *arXiv preprint arXiv:2602.02454*, 2026.
- [9] Y. Shang *et al.*, “Roboscape: Physics-informed embodied world model,” in *NeurIPS*, 2025.
- [10] Z. Liu, X. Deng, S. Chen, A. Wang, Q. Guo, M. Han, Z. Xue, M. Chen, P. Luo, and L. Yang, “Worldweaver: Generating long-horizon video worlds via rich perception,” in *NeurIPS*, 2025.
- [11] S. Gao, S. Zhou, Y. Du, J. Zhang, and C. Gan, “Adaworld: Learning adaptable world models with latent actions,” in *ICML*, 2025.
- [12] M. Assran *et al.*, “V-jepa 2: Self-supervised video models enable understanding, prediction and planning,” *arXiv preprint arXiv:2506.09985*, 2025.
- [13] P. Zhou *et al.*, “ τ_0 -wm: A unified video-action world model for robotic manipulation,” *arXiv preprint arXiv:2606.01027*, 2026.
- [14] Y. Zhang *et al.*, “Imagewam: Do world action models really need video generation, or just image editing?” *arXiv preprint arXiv:2606.19531*, 2026.
- [15] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, “Robocasa: Large-scale simulation of household tasks for generalist robots,” in *RSS*, 2024.
- [16] B. Liu *et al.*, “Libero: Benchmarking knowledge transfer for lifelong robot learning,” in *NeurIPS*, 2023.
- [17] Z. Zhou *et al.*, “Autoeval: Autonomous evaluation of generalist robot manipulation policies in the real world,” in *CoRL*, 2025.
- [18] S. Gao, W. Liang, K. Zheng, A. Malik, S. Ye, S. Yu, W.-C. Tseng, Y. Dong, K. Mo, C.-H. Lin, *et al.*, “Dreamdojo: A generalist robot world model from large-scale human videos,” *arXiv preprint arXiv:2602.06949*, 2026.
- [19] J. Yang *et al.*, “Rise: Self-improving robot policy with compositional world model,” in *RSS*, 2026.
- [20] Z. Zhang *et al.*, “Towards practical world model-based reinforcement learning for vision-language-action models,” in *ICML*, 2026.
- [21] F. Zhu *et al.*, “Wmpo: World model-based policy optimization for vision-language-action models,” in *ICLR*, 2026.
- [22] Y. Guo, L. X. Shi, J. Chen, and C. Finn, “Ctrl-world: A controllable generative world model for robot manipulation,” in *ICLR*, 2026.
- [23] S. Huang, L. Chen, P. Zhou, S. Chen, Y. Liao, Z. Jiang, Y. Hu, P. Gao, H. Li, M. Yao, *et al.*, “Enerverse: Envisioning embodied future space for robotics manipulation,” in *NeurIPS*, 2025.
- [24] S. Zhou, Y. Du, J. Chen, Y. Li, D.-Y. Yeung, and C. Gan, “Robodreamer: Learning compositional world models for robot imagination,” in *ICML*, 2024.
- [25] Q. Bu, Y. Yang, J. Cai, S. Gao, G. Ren, M. Yao, P. Luo, and H. Li, “Learning to act anywhere with task-centric latent actions,” in *RSS*, 2025.
- [26] C. Zhang, T. Pearce, P. Zhang, K. Wang, X. Chen, W. Shen, L. Zhao, and J. Bian, “What do latent action models actually learn?” in *NeurIPS*, 2025.
- [27] S. Ye, J. Jang, B. Jeon, S. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin, *et al.*, “Latent action pretraining from videos,” in *ICLR*, 2025.
- [28] C. Zhang, J. Wang, Z. Gao, Y. Su, T. Dai, C. Zhou, J. Lu, and Y. Tang, “Clap: Contrastive latent action pretraining for learning vision-language-action models from human videos,” *arXiv preprint arXiv:2601.04061*, 2026.
- [29] Z. Yuan *et al.*, “Hermes: Human-to-robot embodied learning from multi-source motion data for mobile dexterous manipulation,” *T-RO*, 2026.
- [30] C. Zhu, R. Yu, S. Feng, B. Burchfiel, P. Shah, and A. Gupta, “Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets,” in *RSS*, 2025.
- [31] R. Zheng, J. Wang, S. Reed, J. Bjorck, Y. Fang, F. Hu, J. Jang, K. Kundalia, Z. Lin, L. Magne, *et al.*, “Flare: Robot learning with implicit world modeling,” in *CoRL*, 2025.
- [32] F. Yang *et al.*, “Chain of world: World model thinking in latent motion,” in *CVPR*, 2026.
- [33] N. Agarwal *et al.*, “Cosmos world foundation model platform for physical ai,” *arXiv preprint arXiv:2501.03575*, 2025.
- [34] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
- [35] T. Chen *et al.*, “A simple framework for contrastive learning of visual representations,” in *ICML*, 2020.
- [36] T. Silver, K. Allen, J. Tenenbaum, and L. Kaelbling, “Residual policy learning,” *arXiv preprint arXiv:1812.06298*, 2018.
- [37] X. Li *et al.*, “Evaluating real-world robot manipulation policies in simulation,” in *CoRL*, 2024.
- [38] Y. Shen, F. Wei, Z. Du, Y. Liang, Y. Lu, J. Yang, N. Zheng, and B. Guo, “Videovla: Video generators can be generalizable robot manipulators,” *NeurIPS*, 2025.
- [39] K. Black *et al.*, “ π_0 : A vision-language-action flow model for general robot control,” *arXiv preprint arXiv:2410.24164*, 2024.
- [40] P. Intelligence *et al.*, “ $\pi_{0.5}$: a vision-language-action model with open-world generalization,” *arXiv preprint arXiv:2504.16054*, 2025.
- [41] S. Li *et al.*, “Unified video action model,” in *RSS*, 2025.
- [42] Y. Wang *et al.*, “Rethinking visual-language-action model scaling: Alignment, mixture, and regularization,” *arXiv preprint arXiv:2602.09722*, 2026.
- [43] M. J. Kim *et al.*, “Openvla: An open-source vision-language-action model,” in *CoRL*, 2024.
- [44] M. J. Kim, C. Finn, and P. Liang, “Fine-tuning vision-language-action models: Optimizing speed and success,” in *RSS*, 2025.
- [45] NVIDIA, “Physicalai-robotics-gr00t-x-embodiment-sim,” 2025, version: be164f2 (commit date: 2025-12-09). License: CC BY 4.0. Accessed: 2026-02-21. [Online]. Available: <https://huggingface.co/datasets/nvidia/PhysicalAI-Robotics-GR00T-X-Embodiment-Sim>
- [46] J. Bjorck *et al.*, “Gr00t n1: An open foundation model for generalist humanoid robots,” *arXiv preprint arXiv:2503.14734*, 2025.
- [47] G. Team. (2025, June) GR00T N1.5: An improved open foundation model for generalist humanoid robots. NVIDIA Research. [Online]. Available: <https://research.nvidia.com/labs/gear/gr00t-n1.5/>
- [48] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel, “Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators,” in *IROS*, 2024.
- [49] S.-W. Lee, X. Kang, B. Y. Yang, and Y.-L. Kuo, “Class: Contrastive learning via action sequence supervision for robot manipulation,” in *CoRL*, 2025.
- [50] N. Saunshi, O. Plevrakis, S. Arora, M. Khodak, and H. Khandeparkar, “A theoretical analysis of contrastive unsupervised representation learning,” in *ICML*, 2019.
- [51] Y. Shirai, K. Ota, D. K. Jha, and D. Romeres, “Learning non-prehensile manipulation with force and vision feedback using optimization-based demonstrations,” *RA-L*, 2026.
- [52] E. Yu *et al.*, “Dm0: An embodied-native vision-language-action model towards physical ai,” *arXiv preprint arXiv:2602.14974*, 2026.