GWM-VLA: GEOMETRY-AWARE LATENT WORLD MODELING FOR VISION-LANGUAGE-ACTION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models achieve strong robotic manipulation performance but often degrade under visual and environmental shifts. Latent world modeling offers a promising approach to improving robustness, yet existing methods commonly encode camera views independently and predict holistic scene dynamics without explicitly modeling their geometric relationships. We propose GWM-VLA, a geometry-aware latent world modeling framework for VLA learning. GWM-VLA combines geometry-aware multi-view state encoding, global context-conditioned target-view prediction, and shared latent-action representations grounded in robot-action supervision. Specifically, VGGT-Ω jointly aggregates multi-view observations at each timestep to construct geometry-aware multi-view states. The latent world model predicts the next-step patch tokens of a selected target view using patch and register tokens extracted after multi-view aggregation, thereby retaining multi-view geometric information without predicting the complete multi-view state. We use the wrist view as the target in our experiments, placing greater emphasis on end-effector motion and local gripper-object interactions. Finally, the shared latent-action representations condition both the latent world model and the flow-matching action head, allowing latent-prediction supervision and ground-truth robot-action supervision to jointly shape the same latent-action representations. Experiments across both simulation and real-world environments demonstrate the effectiveness and robustness of GWM-VLA.

1 INTRODUCTION

Vision-Language-Action (VLA) models (Brohan et al., 2022; Zitkovich et al., 2023) transfer semantic knowledge from large vision-language models to continuous robot control and have achieved strong manipulation performance across diverse tasks. However, even high-performing VLA policies can degrade substantially under controlled changes in camera viewpoint, illumination, background texture, object layout, and sensor noise (Fei et al., 2025). This sensitivity to visual and environmental changes suggests that some policies may rely on appearance-specific correlations rather than capturing how the scene changes in response to robot actions.

Learning environment dynamics through world modeling can improve VLA robustness by encouraging policies to capture action-induced state changes rather than relying mainly on visual appearance. Existing methods incorporate explicit world models in different ways. World-action models such as DreamZero (Ye et al., 2026) integrate future visual prediction and action generation within a unified policy, whereas $\pi_{0.7}$ (Physical Intelligence et al., 2026) uses a separate world model to generate visual subgoals for the VLA. However, these explicit world models require generating future visual observations. Their prediction objectives must therefore model textures, lighting, backgrounds, and other appearance details in addition to control-relevant state changes.

Latent world modeling offers a more compact alternative by predicting future representations rather than generating future visual observations, reducing the need to model fine-grained appearance details. VLA-JEPA (Sun et al., 2026) follows this approach by predicting future visual latents conditioned on learned latent actions. This objective allows the world model to focus on latent state transitions without explicitly generating future images or videos. However, VLA-JEPA uses a pretrained V-

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

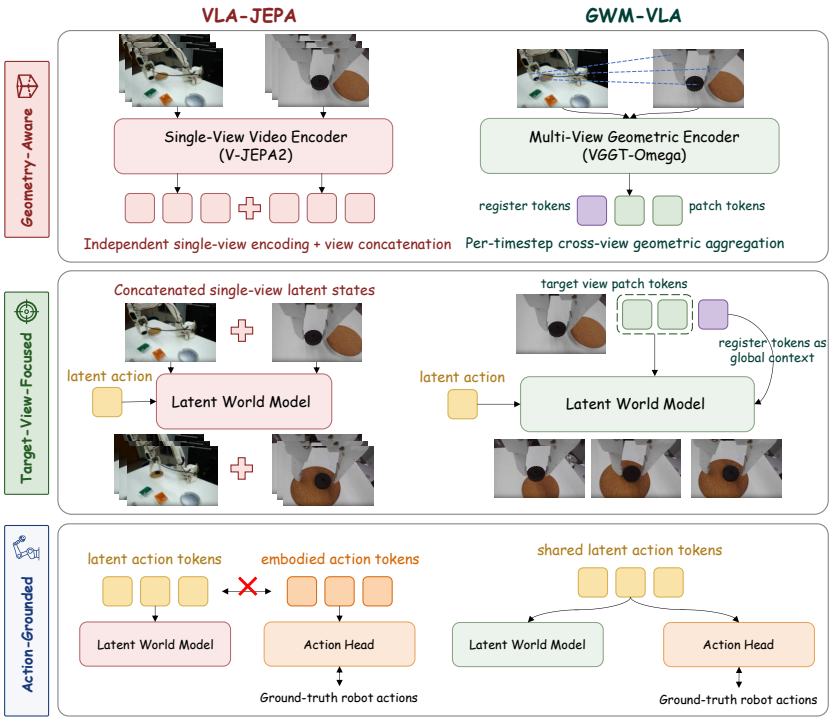

Figure 1: Mechanism-level comparison between VLA-JEPA and GWM-VLA. GWM-VLA combines geometry-aware multi-view state encoding, global context-conditioned target-view prediction, and shared latent-action conditioning.

JEPA2 encoder (Assran et al., 2025) to encode each camera view independently and combines the resulting features through concatenation. Consequently, cross-view geometric relationships, including visual correspondences and camera-relative scene structure, remain implicit.

Recent feed-forward geometric foundation models (Wang et al., 2025; 2026; Leroy et al., 2024; Wang et al., 2024) make geometry-aware multi-view modeling practical. In particular, VGGT (Wang et al., 2025) and VGGT-Ω (Wang et al., 2026) jointly aggregate multiple views and encode cross-view geometric structure, making them attractive visual backbones for spatially grounded robot policies. However, how geometry-aware multi-view representations can serve as predictive states for latent world modeling remains underexplored.

To this end, we propose **GWM-VLA**, a Geometry-Aware Latent World Modeling framework for VLA learning, as illustrated in Figure 1. GWM-VLA introduces three key designs. First, VGGT-Ω aggregates multi-view observations at each timestep to construct geometry-aware multi-view states. Second, rather than predicting the complete multi-view state, the latent world model predicts the next-step patch tokens of a selected target view, while using the corresponding target-view register tokens as global geometric context. This design restricts prediction to the selected view while still leveraging geometric information obtained through multi-view aggregation. We use the wrist view in our experiments, placing greater emphasis on end-effector motion and local gripper-object interactions. Third, the shared latent-action representations condition both the latent world model and the flow-matching action head. Consequently, the latent prediction and robot-action objectives jointly shape the shared latent-action representations, coupling predictive learning with continuous robot control.

We evaluate GWM-VLA in both simulation and real-world settings. On LIBERO, GWM-VLA achieves a 97.1% average success rate, matching the best reported performance while using only robot demonstrations for pretraining. On the more challenging LIBERO-Plus robustness benchmark, GWM-VLA reaches a 75.4% overall success rate, outperforming the robot-only VLA-JEPA baseline by 12.5 percentage points and achieving the highest overall success rate among the compared methods. Real-world experiments on the SO-101 robot further demonstrate that GWM-VLA achieves the highest average success rate under limited demonstrations.

Our main contributions are summarized as follows:

1. We introduce geometry-aware latent world modeling, which uses jointly aggregated multi-view geometric representations as predictive states and models their action-conditioned temporal transitions.
2. We propose GWM-VLA, a VLA framework that integrates geometry-aware latent world modeling with flow-matching action generation through shared latent-action representations.
3. We demonstrate the effectiveness of GWM-VLA for accurate and robust robot control through simulation and real-world experiments.

2 Related Work

Geometry-Aware Representations for VLA. Geometry-aware representations are important for VLA policies because robotic manipulation requires reasoning about object geometry, spatial relationships, camera viewpoints, and gripper-object interactions, beyond recognizing visual semantics alone. Existing methods introduce spatial structure into VLA models through several complementary directions. Spatial Forcing (Li et al., 2025) aligns implicit spatial representations during policy learning, while SpatialVLA (Qu et al., 2025) incorporates spatially structured representations to improve action prediction. Other approaches use explicit 3D information: SUGAR (Chen et al., 2024) pretrains transferable 3D representations for robotic manipulation, Lift3D (Jia et al., 2024) lifts large-scale 2D representations into 3D policy features, and PointVLA (Li et al., 2026) injects point-based scene information into VLA models. 3D-VLA (Zhen et al., 2024) further combines 3D scene understanding with generative world modeling. Feed-forward geometric foundation models such as VGGT and VGGT-Ω (Wang et al., 2025; 2026) provide a way to obtain geometry-aware representations by jointly aggregating multiple views and modeling their cross-view relationships. 3D-Mix (Yu et al., 2026) adaptively fuses VGGT-derived geometry with VLM features, while a recent study systematically examines how geometric foundation-model representations influence VLA action prediction (Yang et al., 2026). These studies demonstrate the value of VGGT-derived geometry for robot control. However, they primarily use geometric representations as perceptual inputs or spatial priors, rather than modeling their temporal evolution through a world-model objective. GWM-VLA instead uses jointly aggregated VGGT-Ω representations as predictive states for action-conditioned latent world modeling.

World Modeling for VLA. World models provide VLA policies with predictive supervision by modeling future observations or representations. This encourages the policy to capture state changes caused by robot actions rather than relying only on the current observation. World-action models (WAMs) integrate visual prediction and robot control within a unified model. DreamZero (Ye et al., 2026), for example, jointly generates future video and robot actions as a closed-loop policy. A recent comparative study reports strong robustness for several WAMs under distribution shifts (Zhang et al., 2026b). World models can also be used as external components for VLA policies. For example, $\pi_{0.7}$ (Physical Intelligence et al., 2026) uses a separate generative world model to produce multi-view subgoal images, which are then provided to the VLA as additional context. Explicit world models generate future images or videos, requiring them to model both task-relevant changes and fine-grained appearance. Latent world models instead predict future representations, providing a more compact objective for policy learning. VLA-JEPA (Sun et al., 2026) follows this approach by predicting future visual latents conditioned on learned latent actions. However, it encodes each camera view independently and combines their features through concatenation, leaving cross-view geometric relationships implicit. It also introduces additional embodied-action representations for action generation rather than directly reusing the latent actions that condition the world model. GWM-VLA instead models next-step transitions over jointly aggregated multi-view geometric states and uses the same latent-action representations for both latent prediction and continuous action generation.

Latent Action Learning. Latent action learning enables policy pretraining from videos without action annotations by representing visual transitions with implicit action variables (Chen et al., 2025b; Bu et al., 2025a). LAPA (Ye et al., 2025) learns discrete latent actions for VLA pretraining, while UniVLA (Bu et al., 2025b) introduces language-guided, task-centric latent actions. Villa-X (Chen et al., 2025a) further grounds latent actions in physical dynamics by predicting proprioceptive states and jointly modeling latent and robot actions. However, transition-based objectives may capture

dominant visual changes, such as camera motion and background variation, rather than changes caused by robot control alone (Zhang et al., 2026a; Lin et al., 2026; Nikulin et al., 2025). MVP-LAM (Lee et al., 2026) reduces viewpoint dependence through cross-view reconstruction. These methods primarily focus on action representation learning rather than modeling the future evolution of latent environment states.

3 METHODOLOGY

3.1 OVERVIEW

We consider a robot manipulation trajectory segment of horizon T consisting of multi-view observations, a language instruction, proprioceptive states, and continuous robot actions:

$$\boldsymbol{\tau} = (\mathbf{O}_{0:T}, \ell, \mathbf{s}_{0:T}, \mathbf{u}_{0:T-1}) \tag{1}$$

where T denotes the shared prediction and action horizon, and $\mathbf{O}_t$ denotes the multi-view observation:

$$\mathbf{O}_t = \{I_t^v\}_{v=1}^V \tag{2}$$

Here ℓ denotes the language instruction, $\mathbf{s}_t$ is the robot proprioceptive state, and $\mathbf{u}_t$ is the executed action. As illustrated in Fig. 2, **GWM-VLA** consists of three main components: (1) a geometry-aware multi-view encoder based on VGGT-Ω, (2) a latent world model with global context-conditioned target-view prediction, and (3) a flow-matching action policy conditioned on a unified latent action representation.

GWM-VLA jointly encodes multi-view observations at each timestep with VGGT-Ω, while processing different timesteps independently. The latent world model performs causal, teacher-forced next-step prediction over the geometry-aware latent sequence, while the VLM-derived latent-action representation conditions both the latent world model and the flow-matching policy.

3.2 GEOMETRY-AWARE MULTI-VIEW STATE ENCODING

Generic visual encoders are not explicitly optimized to model the geometric relationships shared across multi-view observations. In contrast, robotic manipulation requires spatial reasoning about object geometry, camera geometry, and gripper-object interactions. We therefore employ a pretrained VGGT-Ω encoder to construct geometry-aware multi-view states. The VGGT-Ω aggregator is frozen during training.

At each timestep t, multi-view observations are jointly processed as a single geometric sample:

$$\{\mathbf{R}_t, \mathbf{P}_t\} = E_\Omega\left(\{I_t^v\}_{v=1}^V\right), \tag{3}$$

where E_Ω denotes the VGGT-Ω encoder, and $\mathbf{R}_t$ and $\mathbf{P}_t$ denote the register and patch tokens, respectively. Equation 3 is applied independently at each timestep, while all views at the same timestep are jointly aggregated by the geometry encoder.

3.3 GLOBAL CONTEXT-CONDITIONED TARGET-VIEW PREDICTION

We perform next-step prediction in the geometry-aware latent space of VGGT-Ω rather than reconstructing future RGB observations. Predicting the complete multi-view state would distribute the prediction objective across all camera streams. We therefore apply the prediction loss to a selected target view and condition the latent world model on the corresponding target-view register tokens. Since both the patch and register tokens are extracted after joint multi-view aggregation, the model predicts only one view while still leveraging geometric information from multi-view observations.

Let $v^\star$ denote a target view selected according to the sensor configuration and task distribution. After multi-view aggregation, we extract the corresponding register and patch tokens:

$$\mathbf{r}_t = \mathbf{R}_t^{v^\star}, \qquad \mathbf{p}_t = \mathbf{P}_t^{v^\star}. \tag{4}$$

Here, $\mathbf{p}_t$ defines the target-view latent state to be predicted, while $\mathbf{r}_t$ provides global geometric context to the latent world model. Both token sets correspond to the selected view but are produced

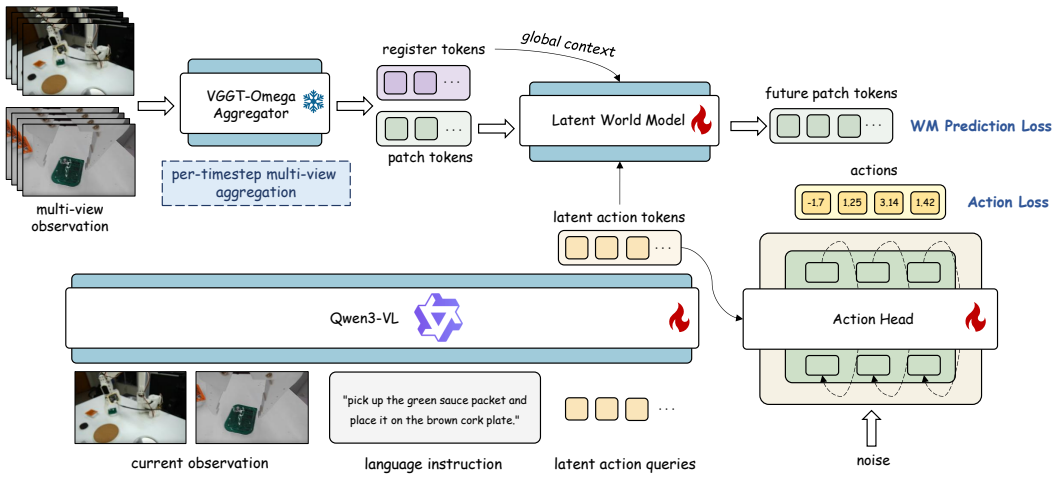

Figure 2: Overall architecture of GWM-VLA. A frozen VGGT-Ω encoder aggregates multi-view observations at each timestep. Shared latent-action tokens condition both next-step latent prediction and flow-matching action generation. Snowflakes denote frozen modules, and flames denote trainable modules.

after cross-view interaction within VGGT-Ω. In our experiments, we select the wrist view because its observations are closely coupled with end-effector motion and local gripper-object interactions. We further condition the latent world model on a latent-action sequence $\mathbf{A}_{0:T-1}$ produced by the vision-language backbone and shared with the action policy. Given the target-view patch tokens, target-view register tokens, and latent action tokens, the latent world model estimates the next target-view patch tokens at each timestep:

$$\hat{\mathbf{p}}_{1:T} = F_\phi\left(\mathbf{p}_{0:T-1}, \mathbf{r}_{0:T-1}, \mathbf{A}_{0:T-1}\right), \tag{5}$$

where F_ϕ denotes the action-conditioned next-step latent world model.

The latent world model employs time-causal attention. Tokens within the same timestep can interact bidirectionally, while tokens at timestep t can attend only to tokens from timesteps no later than t. The attention mask is defined as

$$M_{ij} = \begin{cases} 0, & \tau(j) \le \tau(i), \\ -\infty, & \tau(j) > \tau(i), \end{cases} \tag{6}$$

where $\tau(i)$ denotes the timestep associated with token i.

During training, we adopt teacher forcing. The ground-truth latent state $\mathbf{p}_t$ is provided when predicting the next state $\mathbf{p}_{t+1}$. The supervision targets are obtained by encoding future multi-view observations with the same per-timestep geometry encoder:

$$\mathbf{p}_{t+1} = E_\Omega^{P,v^\star}\left(\mathbf{O}_{t+1}\right), \tag{7}$$

where $E_\Omega^{P,v^\star}$ denotes the target-view patch-token output of VGGT-Ω. Together with the causal attention mask in the latent world model, this yields an action-conditioned prediction objective over the geometry-aware states defined above.

We optimize the world model using an ℓ_1 latent prediction objective:

$$\mathcal{L}_{\text{wm}} = \frac{1}{T}\sum_{t=0}^{T-1} \left\|\hat{\mathbf{p}}_{t+1} - \mathbf{p}_{t+1}\right\|_1, \tag{8}$$

3.4 Unified Latent Action Representation

To connect latent dynamics learning with robot control, GWM-VLA uses a shared latent-action representation to condition both the world model and the action policy. We adopt Qwen3-VL-2B (Bai et al., 2025) as the vision-language backbone. Given the current multi-view observation and the

language instruction, we insert timestep-grouped learnable latent action query tokens:

$$\begin{aligned} \mathcal{Q}^A &= \left[\mathcal{Q}_0^A, \mathcal{Q}_1^A, \ldots, \mathcal{Q}_{T-1}^A\right], \\ \mathcal{Q}_t^A &= \left[q_{t,1}^A, \ldots, q_{t,K}^A\right]. \end{aligned} \tag{9}$$

All timestep-grouped latent-action representations are produced from the current observation $\mathbf{O}_0$. The index t denotes a position in the future action horizon rather than a separately observed policy input. The horizon T is shared by latent prediction and action generation, so each predicted transition and action-chunk position has one timestep-indexed latent-action group.

The corresponding latent action representations are obtained from the hidden states of the vision-language model:

$$\mathbf{A}_{0:T-1} = Q_\theta\left(\mathbf{O}_0, \ell, \mathcal{Q}^A\right), \tag{10}$$

where

$$\mathbf{A}_t = [\mathbf{a}_{t,1}, \ldots, \mathbf{a}_{t,K}] \tag{11}$$

contains K latent action tokens associated with timestep t. Unlike VLA-JEPA, which introduces additional embodied action queries for action generation, GWM-VLA directly reuses the latent-action sequence $\mathbf{A}_{0:T-1}$ to condition both the latent world model and the flow-matching action head. This shared design allows the latent prediction and robot-action objectives to jointly shape the same latent-action representations, encouraging them to capture dynamics relevant to robot control.

3.5 Conditional Flow-Matching Action Head

To capture the multimodal distribution of continuous robot actions, we employ a conditional flow-matching head (Lipman et al., 2023) to generate action chunks. The action head is conditioned on the shared latent-action sequence $\mathbf{A}_{0:T-1}$ and the current proprioceptive state $\mathbf{s}_0$, and generates an action chunk over horizon T. Given a ground-truth action chunk $\mathbf{u}_{0:T-1}$, Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, flow time $\gamma \sim \mathcal{U}(0,1)$, and $\mathbf{u}_\gamma = (1-\gamma)\boldsymbol{\epsilon} + \gamma \mathbf{u}_{0:T-1}$, we optimize

$$\mathcal{L}_{\text{action}} = \mathbb{E}\Big[\left\| v_\psi\left(\mathbf{u}_\gamma, \gamma \mid \mathbf{A}_{0:T-1}, \mathbf{s}_0\right) - \left(\mathbf{u}_{0:T-1} - \boldsymbol{\epsilon}\right)\right\|_2^2\Big]. \tag{12}$$

Full training and inference details are provided in Appendix A.

3.6 Joint World-Model and Policy Learning

GWM-VLA jointly optimizes teacher-forced next-step latent prediction and robot action generation:

$$\mathcal{L} = \mathcal{L}_{\text{action}} + \lambda \mathcal{L}_{\text{wm}}, \tag{13}$$

where λ balances the predictive and control objectives. We use $\lambda = 0.1$ as the default world-model loss weight in the main experiments; a sensitivity analysis is provided in Appendix B. This joint objective encourages the latent-action representations to encode action-conditioned interaction changes while remaining grounded in executable robot controls. The world-model loss serves as an auxiliary training objective and is not used during deployment.

4 Experiments

We evaluate GWM-VLA in simulation and on a real robot, including robustness benchmarks, controlled ablations, and qualitative probing of predicted latent states.

4.1 Experimental Setup

Implementation details. GWM-VLA uses Qwen3-VL-2B as the vision-language backbone and VGGT-Ω as the geometry-aware multi-view encoder. The VGGT-Ω aggregator is frozen throughout training, while Qwen3-VL, the latent world model, and the action head are fully trainable. Unless otherwise specified, we use the final VGGT-Ω representation, set the world-model loss weight to $\lambda = 0.1$, and directly condition the action head on the shared latent-action tokens. Pretraining and simulation fine-tuning use 8 NVIDIA A800 GPUs, while ablation experiments use a single

NVIDIA RTX 6000D GPU. Additional training details and hyperparameter analyses are provided in Appendix A.

Datasets. We pretrain GWM-VLA on DROID (Khazatsky et al., 2024), a large-scale action-labeled robot dataset containing approximately 76,000 demonstration trajectories collected across multiple institutions and operators. The dataset covers diverse manipulation tasks, camera viewpoints, workspace layouts, and visual backgrounds, providing broad variation for learning transferable robot representations. For simulation experiments, we fine-tune the pretrained model on the standard demonstration sets provided by LIBERO.

Benchmarks. We conduct simulation experiments on the LIBERO (Liu et al., 2023) and LIBERO-Plus (Fei et al., 2025) benchmarks. LIBERO uses the Franka Emika Panda arm and includes four task suites, SPATIAL, OBJECT, GOAL, and LIBERO-10, which evaluate spatial reasoning, object-centric manipulation, goal-conditioned control, and long-horizon execution. We use it as the in-distribution simulation benchmark. LIBERO-Plus extends the same task families across seven perturbation dimensions, including camera viewpoint, robot initial state, language instruction, illumination, background, visual noise, and object layout, and serves as our out-of-distribution robustness benchmark.

Baselines. We compare GWM-VLA with representative VLA, latent-action, and world-model-based methods, including LAPA (Ye et al., 2025), UniVLA (Bu et al., 2025b), OpenVLA-OFT (Kim et al., 2025), π_0 (Black et al., 2024), π_0-FAST (Pertsch et al., 2025), CoT-VLA (Zhao et al., 2025), WorldVLA (Cen et al., 2025), villa-X (Chen et al., 2025a), GR00T N1 (Bjorck et al., 2025), $\pi_{0.5}$ (Intelligence et al., 2025), and VLA-JEPA (Sun et al., 2026). Because GWM-VLA is pretrained only on robot demonstrations, robot-only VLA-JEPA without human-video pretraining serves as the most directly comparable baseline.

Table 1: LIBERO success rates (%). The best and runner-up results are highlighted.

Method	Spatial	Object	Goal	LIBERO-10	Avg.
LAPA (Ye et al., 2025)	73.8	74.6	58.8	55.4	65.7
UniVLA (Bu et al., 2025b)	96.5	96.8	95.6	92.0	95.2
OpenVLA-OFT (Kim et al., 2025)	97.6	98.4	97.9	**94.5**	**97.1**
π_0 (Black et al., 2024)	96.8	98.8	95.8	85.2	94.2
π_0-FAST (Pertsch et al., 2025)	96.4	96.8	88.6	60.2	85.5
CoT-VLA (Zhao et al., 2025)	87.5	91.6	87.6	69.0	83.9
WorldVLA (Cen et al., 2025)	87.6	96.2	83.4	60.0	81.8
villa-X (Chen et al., 2025a)	97.5	97.0	91.5	74.5	90.1
GR00T N1 (Bjorck et al., 2025)	94.4	97.6	93.0	90.6	93.9
$\pi_{0.5}$ (Intelligence et al., 2025)	**98.8**	98.2	**98.0**	92.4	96.9
VLA-JEPA (Sun et al., 2026) w/o human videos	94.8	**99.6**	95.8	94.0	96.1
GWM-VLA (ours)	96.8	99.0	**98.0**	94.4	**97.1**

Table 2: LIBERO-Plus success rates (%). The best and runner-up results are highlighted.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Avg.
UniVLA (Bu et al., 2025b)	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
OpenVLA-OFT (Kim et al., 2025)	56.4	31.9	79.5	88.7	**93.3**	75.8	74.2	69.6
π_0 (Black et al., 2024)	13.8	6.0	58.8	85.0	81.4	**79.0**	68.9	53.6
π_0-FAST (Pertsch et al., 2025)	**65.1**	21.6	61.0	73.2	73.2	74.4	68.8	61.6
WorldVLA (Cen et al., 2025)	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
VLA-JEPA (Sun et al., 2026) w/o human videos	40.3	**55.7**	72.9	88.2	70.5	38.2	74.6	62.9
GWM-VLA (ours)	57.9	54.7	**89.8**	**95.4**	90.8	72.5	**77.1**	**75.4**

4.2 SIMULATION BENCHMARK EXPERIMENTS

LIBERO. As shown in Table 1, GWM-VLA achieves a 97.1% average success rate, tying OpenVLA-OFT for the best overall performance and outperforming the robot-only VLA-JEPA baseline by 1.0 percentage point. Its consistently strong performance across all four task suites shows that geometry-aware latent world modeling preserves effective in-distribution control while introducing additional geometric and predictive supervision.

LIBERO-Plus. Table 2 reports the results on LIBERO-Plus. GWM-VLA achieves the highest overall success rate among the compared methods at 75.4%, exceeding the robot-only VLA-JEPA baseline by 12.5 percentage points. It ranks first under language, lighting, and object-layout perturbations, and second under camera-viewpoint, robot-initial-state, and background perturbations. The largest gains over VLA-JEPA occur under visual noise, background, camera-viewpoint, and language changes. This broad improvement is consistent with jointly aggregated geometric states providing more stable spatial representations under visual and environmental shifts.

4.3 Real-World Experiments

Setup. We further evaluate GWM-VLA on the SO-101 robot using the open-source LeRobot deployment stack (Cadene et al., 2026). We collect 100 teleoperated robot demonstrations over five training pick-and-place tasks, and all methods are adapted using this same demonstration set. Each evaluation task is attempted 10 times. We consider three settings: in-distribution (ID), consisting of three seen instructions from the training task set under nominal layouts; OOD task, consisting of three held-out object–receptacle recombinations; and OOD layout, consisting of the same three ID instructions evaluated under novel object arrangements. We compare against π_0 (Black et al., 2024) and $\pi_{0.5}$ (Intelligence et al., 2025).

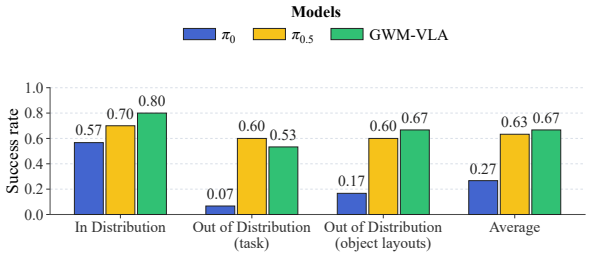

Figure 3: **Real-world experimental results.** Success rates under ID, OOD-task, and OOD-layout settings.

Results. Figure 3 summarizes the real-world results. GWM-VLA obtains the highest observed average success rate in our real-world evaluation and is most effective in the ID and OOD-layout settings. These results are consistent with geometry-aware state prediction providing a useful inductive bias for execution accuracy and spatial robustness under limited real-world demonstrations. The OOD-task setting presents a different challenge: the policy must recombine object and receptacle concepts into held-out instructions. Here $\pi_{0.5}$ performs best at 60.0%, while GWM-VLA reaches 53.3% and substantially outperforms π_0. This result suggests that spatial robustness and compositional instruction generalization remain distinct challenges.

4.4 Ablation Studies

We conduct ablations on LIBERO (spatial suite) to investigate the key design choices of GWM-VLA. The default configuration uses the final VGGT-Ω layer and direct latent-action conditioning. These ablations are lightweight diagnostic runs trained only on LIBERO (spatial suite) using a single NVIDIA RTX 6000D GPU, with a batch size of 16 and 10,000 optimization steps. They are therefore intended to compare design variants under the same restricted setting, rather than to be directly compared with the main pretraining-plus-fine-tuning results.

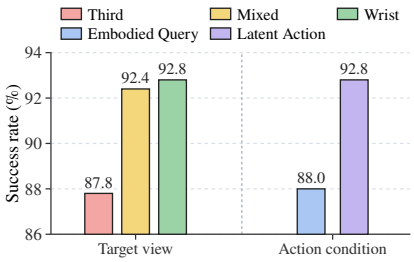

Figure 4: **Diagnostic ablations.** Target-view selection (left) and action conditioning (right) on LIBERO (spatial suite).

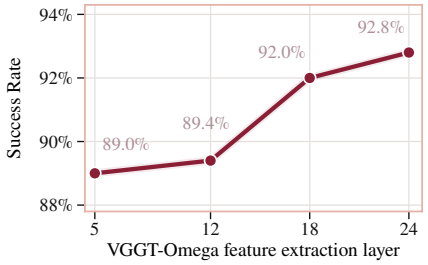

Figure 5: **Representation depth.** Effect of VGGT-Ω representation depth on LIBERO (spatial suite).

Effect of Target-View Selection. We study how the choice of target view affects policy performance in the LIBERO (spatial suite) setting. As shown in Figure 4 (left), wrist-view prediction achieves a

92.8% success rate, outperforming third-person-view prediction at 87.8%. Mixed-view prediction, which samples 80% wrist-view targets and 20% third-person-view targets, achieves a comparable but slightly lower success rate of 92.4%. These results suggest that the wrist view is the most effective target among the evaluated configurations. We attribute the advantage of the wrist view to its close relation to end-effector motion and local gripper-object interactions. Because these tokens are extracted after joint multi-view aggregation, the prediction target retains global geometric context while emphasizing local manipulation dynamics.

Effect of the Shared Latent-Action Representation. We compare direct conditioning of the action head on latent-action tokens with conditioning on an additional embodied-action query. The action-conditioning results are shown in Figure 4 (right). Under the controlled LIBERO (spatial suite) setting, direct latent-action conditioning outperforms the embodied-query variant by 4.8 percentage points. This result supports directly sharing the latent-action tokens between latent prediction and action generation.

Effect of the Representation Depth. The representation depth here refers to the VGGT-Ω layer from which the patch and register tokens are extracted to construct the latent state. Prior studies report substantial layer-wise differences in VGGT: cross-view correspondences and epipolar geometry emerge in middle layers and are refined in later layers (Bratulić et al., 2026; You et al., 2026). We therefore compare VGGT-Ω representations extracted from Layers 5, 12, 18, and 24. As shown in Figure 5, the success rate increases with representation depth, and Layer 24 performs best. This result suggests that latent world modeling benefits from the more refined geometric representations available in later VGGT-Ω layers. We therefore use Layer 24 by default.

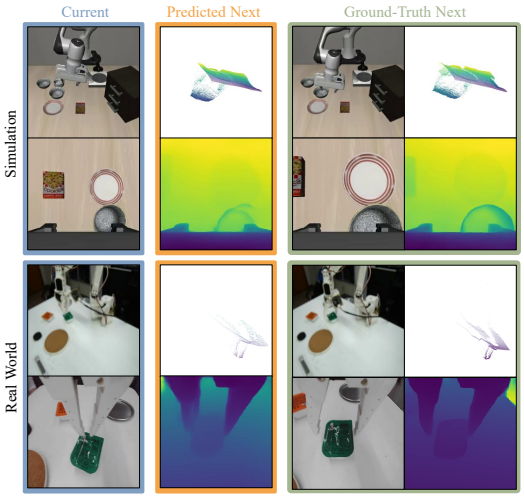

Figure 6: **Predicted latent states.** Qualitative probing of predicted next-step latent states using a separately trained and frozen depth decoder and camera-coordinate point clouds.

Qualitative Probing. Figure 6 qualitatively probes whether predicted latent states retain next-step geometric information. We train a visualization decoder to map observed next-step VGGT-Ω tokens to depth maps produced by the frozen dense head. We then freeze this decoder and apply it to predicted tokens. The decoded results preserve the main scene layout and geometry around the gripper and manipulated objects, although fine details are smoothed. These results are qualitative probes rather than measurements of 3D reconstruction accuracy.

5 Conclusion and Limitations

We propose GWM-VLA, a geometry-aware latent world modeling framework for Vision-Language-Action learning. GWM-VLA jointly aggregates multi-view observations at each timestep with VGGT-Ω to construct geometry-aware states, predicts next-step target-view patch tokens using the corresponding register tokens as global geometric context, and uses shared latent-action representations for both latent prediction and continuous action generation. Across LIBERO and LIBERO-Plus, GWM-VLA achieves strong in-distribution performance and substantial robustness improvements under visual and environmental shifts. Real-world experiments further demonstrate its effectiveness under limited demonstrations and novel object layouts. Despite these results, GWM-VLA relies on multi-view observations at the same timestep to construct geometry-aware latent states, which limits its direct applicability to large-scale human-video pretraining, where such multi-view observations are often unavailable. It also uses a fixed target view during training. **Limitations.** Although the wrist view performs best on the LIBERO spatial suite, the optimal target may vary across tasks and sensor configurations. Future work will extend geometry-aware latent world modeling to single-view video data and explore adaptive target-view selection and multi-target prediction.

AI USE STATEMENT

Generative AI tools were used for language polishing, proofreading, formatting assistance, and routine editing of the presentation of tables and figures. They were not used to generate the research ideas, design the method or experiments, conduct the experiments, analyze the data, or make scientific decisions. All AI-assisted outputs were manually checked and revised by the authors. The authors take full responsibility for the final content of this work, including all methods, claims, and reported results.

ETHICS STATEMENT

This work uses the publicly available DROID, LIBERO, and LIBERO-Plus robot-learning resources, together with robot demonstrations collected by the authors for the SO-101 evaluation. The experiments do not involve human subjects as research participants or the collection of personal or sensitive user data. The real-robot experiments are conducted in controlled laboratory settings, and the authors follow the applicable dataset, software, and data-use requirements.

REPRODUCIBILITY STATEMENT

The proposed architecture and training objectives are described in Sections 3.1–3.6, while the main benchmark settings and evaluation protocols are given in Sections 4.1–4.3. Appendix A provides the model configuration, training schedules, hardware, baseline details, evaluation rules, and additional ablation and real-world evaluation protocols. Together, these sections document the implementation and experimental choices needed to reproduce the reported results; any code provided with the supplementary materials should be used alongside them.

REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

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.

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

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

Jelena Bratulić, Sudhanshu Mittal, Thomas Brox, and Christian Rupprecht. On geometric understanding and learned priors in feed-forward 3d reconstruction models, 2026. URL https://arxiv.org/abs/2512.11508.

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

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025a.

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

Remi Cadene, Simon Alibert, Francesco Capuano, Michel Aractingi, Adil Zouitine, Pepijn Kooijmans, Jade Choghari, Martino Russi, Caroline Pascal, Steven Palma, Mustafa Shukor, Jess Moss, Alexander Soare, Dana Aubakirova, Quentin Lhoest, Quentin Gallouédec, and Thomas Wolf. Lerobot: An open-source library for end-to-end robot learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.22818.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Shizhe Chen, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Sugar: Pre-training 3d visual representations for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18049–18060, 2024.

Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, et al. Villa-x: enhancing latent action modeling in vision-language-action models. *arXiv preprint arXiv:2507.23682*, 2025a.

Yi Chen, Yuying Ge, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for robot manipulation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2025b.

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

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

Yueru Jia, Jiaming Liu, Sixiang Chen, Chenyang Gu, Zhilue Wang, Longzan Luo, Lily Lee, Pengwei Wang, Zhongyuan Wang, Renrui Zhang, et al. Lift3d foundation policy: Lifting 2d large-scale pretrained models for robust 3d robotic manipulation. *arXiv preprint arXiv:2411.18623*, 2024.

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

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

Jung Min Lee, Dohyeok Lee, Seokhun Ju, Taehyun Cho, Jin Woo Koo, Li Zhao, Sangwoo Hong, and Jungwoo Lee. Mvp-lam: Learning action-centric latent action via cross-viewpoint reconstruction. *arXiv preprint arXiv:2602.03668*, 2026.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pp. 71–91. Springer, 2024.

Chengmeng Li, Junjie Wen, Yaxin Peng, Yan Peng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models. *IEEE Robotics and Automation Letters*, 11(3):2506–2513, 2026.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. *arXiv preprint arXiv:2510.12276*, 2025.

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

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

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. *arXiv preprint arXiv:2502.00379*, 2025.

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

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.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model, 2025. URL https://arxiv.org/abs/2501.15830.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. Vggt-ω. *arXiv preprint arXiv:2605.15195*, 2026.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20697–20709, 2024.

Yurou Yang, Muyuan Lin, Roberto Martin-Martin, Martin Labrie, Shreekant Gayaka, Cheng-Hao Kuo, and Luca Carlone. Understanding the impact of geometric foundation models on vision-language-action models. *arXiv preprint arXiv:2605.24642*, 2026.

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

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

Jinhao You, Shuo Lyu, Zhuohang Lyu, Tanxuan Li, Zibo Zhao, Jiaxiang Hu, Kai Tang, and Yichen Guo. Regimevggt: Layer-wise spatially preserving redundancy removal for visual geometry grounded transformer. *arXiv preprint arXiv:2606.18439*, 2026.

Bin Yu, Shijie Lian, Xiaopeng Lin, Zhaolong Shen, Yuliang Wei, Haishan Liu, Changti Wu, Hang Yuan, Bailing Wang, Cong Huang, et al. 3d-mix for vla: A plug-and-play module for integrating vggt-based 3d information into vision-language-action models. *arXiv preprint arXiv:2603.24393*, 2026.

Chuheng Zhang, Tim Pearce, Pushi Zhang, Kaixin Wang, Xiaoyu Chen, Wei Shen, Li Zhao, and Jiang Bian. What do latent action models actually learn? *Advances in Neural Information Processing Systems*, 38:146676–146697, 2026a.

Zhanguang Zhang, Zhiyuan Li, Behnam Rahmati, Rui Heng Yang, Yintao Ma, Amir Rasouli, Sajjad Pakdamansavoji, Yangzheng Wu, Lingfeng Zhang, Tongtong Cao, et al. Do world action models generalize better than vlas? a robustness study. *arXiv preprint arXiv:2603.22078*, 2026b.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1702–1713, 2025.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model, 2024. URL https://arxiv.org/abs/2403.09631.

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

A Experimental Details

A.1 Model Architecture

We use Qwen3-VL-2B as the vision-language backbone. Latent-action tokens are implemented as learnable query embeddings inserted into the Qwen3-VL input. The default configuration uses $K = 8$ latent-action tokens for each of the $T = 7$ future positions shared by transition prediction and action generation, yielding 56 latent-action tokens per sample. These same latent-action tokens condition both the latent world model and the flow-matching action head. We do not use an additional embodied-action query in the default model; the embodied-query variant is evaluated only in the ablation study.

Both DROID pretraining and LIBERO fine-tuning use two views for geometry-aware state encoding: one main view and one wrist view. For DROID, the main view is selected from an exterior camera stream and paired with the wrist camera. For LIBERO, the primary image is used as the global view and the wrist image is used as the target view. The implementation uses $T + 1 = 8$ observation frames during teacher-forced latent prediction, corresponding to $T = 7$ predicted latent transitions and a 7-step action chunk.

Observation images used by Qwen3-VL are resized to 224×224, while images used by the VGGT-Ω geometry encoder are resized to 256×256. VGGT-Ω is applied independently at each timestep to avoid temporal information leakage, but all views within the same timestep are jointly aggregated.

For continuous action prediction, we use a DiT-B Transformer action head trained with flow matching. The action head is conditioned on the latent-action-token sequence extracted from Qwen3-VL and the current robot proprioceptive state. During inference, the action head generates actions through iterative flow-matching denoising. At deployment, both VGGT-Ω and the latent world model are removed, and action generation uses only Qwen3-VL and the flow-matching action head.

In both DROID pretraining and LIBERO fine-tuning, the action head predicts a 7-step future action trajectory, with each action having 7 dimensions. We use an 8-dimensional proprioceptive state. The DiT-B action head uses 16 Transformer layers and 12 attention heads. It contains 32 learnable future query tokens and uses learnable positional embeddings.

Table 3: Configuration of the Latent World Model.

Parameter	**Value**
Geometry encoder	VGGT-Ω
Encoder input resolution	256×256
Encoder patch size	16
Target view	Wrist view
Number of views	2
Observation frames per sample	8
Prediction horizon T	7
Patch tokens per timestep	256
Register tokens per timestep	16
Patch/register token dimension	1024
Latent-action token dimension	2048
Action tokens per transition	8
Predictor Transformer layers	12
Predictor attention heads	8
Predictor attention type	Time-causal
World-model loss	ℓ_1 latent prediction

A.2 Training Details

We pretrain GWM-VLA on DROID for $50{,}000$ optimization steps using 8 NVIDIA A800 GPUs. The per-device batch size is 16 and gradient accumulation is 2, giving an effective global batch size of $16 \times 2 \times N_{\mathrm{GPU}}$, where N_{GPU} is the number of distributed training processes. Training on 8 GPUs therefore gives a global batch size of 256.

Table 4: Configuration of the Action Head.

Parameter	Value
Action head type	Flow matching DiT-B
Transformer layers	16
Attention heads	12
Internal token dimension	768
Output hidden dimension	1024
Condition dimension	2048
State dimension	8
Action dimension	7
Future action horizon	7
Future query tokens	32
Positional encoding	Learnable
Inference denoising steps	4
Training diffusion repeats	8

For simulation fine-tuning, we train on all LIBERO tasks jointly for $30{,}000$ optimization steps using 8 NVIDIA A800 GPUs. The per-device batch size is 16 and gradient accumulation is 2, giving the same effective global batch size of 256 on 8 GPUs. All ablation experiments are conducted on LIBERO (spatial suite) using a single NVIDIA RTX 6000D GPU, with a batch size of 16 and $10{,}000$ optimization steps.

For both DROID pretraining and LIBERO fine-tuning, we optimize the action loss and latent world-model loss jointly, $\mathcal{L} = \mathcal{L}_{\text{action}} + \lambda \mathcal{L}_{\text{wm}}$. We use $\lambda = 0.1$ in the main experiments.

We use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$, and weight decay 10^{-8}. We adopt a cosine learning-rate schedule with a minimum learning rate of 10^{-6} and 5000 warmup steps. The peak learning rates are 1×10^{-5} for the Qwen3-VL backbone and 1×10^{-4} for the action head, with a base learning rate of 3×10^{-5}. Gradient clipping is applied with a maximum norm of 1.0. We use bfloat16 mixed-precision training and DeepSpeed/Accelerate for distributed optimization.

Table 5: Training schedule for DROID pretraining and LIBERO simulation fine-tuning.

Stage	Data	Steps	Batch	Accum.
Pretraining	DROID	50K	16	2
Fine-tuning	LIBERO	30K	16	2

A.3 VLA Baselines and Evaluation Protocols

Prior baselines. Results for prior VLA methods in the simulation tables are sourced from the corresponding papers or their reported benchmark tables, with overall success rates following the aggregation rules described below; we do not re-implement or re-evaluate these simulation baselines. For the real-world comparison, we evaluate the listed generalist VLA baselines under the physical setup and task protocol described in the real-world evaluation section below. Depending on the benchmark, the comparison uses the applicable subset of the following prior methods.

- **OpenVLA-OFT**: an optimized fine-tuning recipe built on OpenVLA with parallel decoding, action chunking, continuous actions, and an ℓ_1 objective.
- π_0: a generalist VLA using a flow-matching action model trained on large-scale multi-robot data.
- π_0**-FAST**: a π_0 variant with a DCT-based discrete action tokenizer for continuous action chunks.
- $\pi_{0.5}$: an extension of π_0 with heterogeneous co-training for open-world generalization.
- **LAPA**: a latent-action pretraining method that learns discrete latent actions from videos before policy fine-tuning.
- **UniVLA**: a latent-action VLA with language-conditioned task-centric latent actions.
- **CoT-VLA**: a VLA that predicts future visual goals as visual chain-of-thought plans before action generation.
- **WorldVLA**: a world-model-based VLA that couples policy learning with an autoregressive action world model.

- **villa-X**: a latent-action VLA that jointly models latent and robot actions.
- **GR00T N1**: a dual-system VLA with a vision-language context module and a diffusion Transformer action module.
- **VLA-JEPA**: a latent-world-model VLA that learns latent actions through future visual-latent prediction; we use its robot-only variant without human-video pretraining.

LIBERO. For LIBERO, we report task success rates under the standard LIBERO evaluation setting. Each task is evaluated with 50 rollouts, and we report success rates for the SPATIAL, OBJECT, GOAL, and LIBERO-10 suites. The four suites have equal numbers of evaluation rollouts, so the overall success rate is the arithmetic mean of the four suite success rates.

LIBERO-Plus. For LIBERO-Plus, we report task success rates across seven robustness perturbation dimensions: camera viewpoint, robot initial state, language instruction, illumination, background, visual noise, and object layout. Evaluation rollout counts differ across these perturbation categories. Let N_d and S_d denote the numbers of evaluated and successful rollouts, respectively, in category d. The overall success rate reported in the **Avg.** column is

$$\mathrm{SR}_{\mathrm{overall}} = 100\% \times \frac{\sum_{d=1}^{7} S_d}{\sum_{d=1}^{7} N_d}. \tag{14}$$

Thus, each category contributes in proportion to its evaluation rollout count; the seven displayed category success rates are not weighted equally.

B Additional Experimental Results

B.1 World-Model Loss Weight Ablation

We assess the effect of the world-model loss by varying its weight λ in the joint training objective and evaluating each setting on LIBERO (spatial suite). Here, $\lambda = 0$ removes the world-model loss and therefore serves as a direct control without latent world-model supervision. As shown in Figure 7, the no-world-model-loss control achieves an 88.0% success rate, while introducing a small auxiliary weight of $\lambda = 0.01$ improves performance to 90.6%. Increasing the weight to $\lambda = 0.1$ further improves the success rate to 92.8%, the best result among the tested settings and a 4.8 percentage-point improvement over $\lambda = 0$. A larger weight of $\lambda = 1.0$ achieves 91.0%, remaining above the no-world-model-loss control but below $\lambda = 0.1$. This trend suggests that moderate world-model supervision provides a useful auxiliary signal, whereas placing too much emphasis on latent prediction can reduce the benefit for action learning. We therefore use $\lambda = 0.1$ in the main experiments.

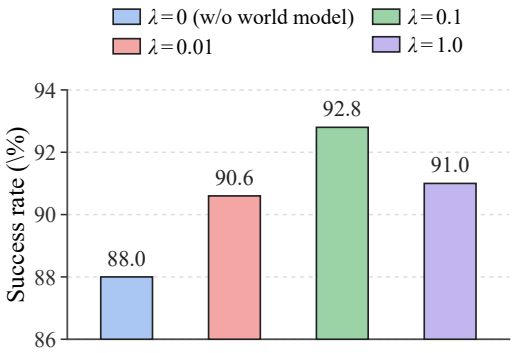

Figure 7: **World-model loss weight.** Ablation of the world-model loss weight on LIBERO (spatial suite).

B.2 Multi-view Aggregation Ablation

We ablate multi-view geometric aggregation in the latent world model on LIBERO (spatial suite). The wrist-only variant feeds only the wrist-view image to VGGT-Ω at each timestep, so the latent world model is supervised by a single-view geometry-aware state rather than the state produced from aggregated multi-view observations. As shown in Figure 8, this variant decreases the success rate from 92.8% to 91.8%, indicating that multi-view geometry provides complementary scene-level context for policy learning.

B.3 Register-Token Conditioning Ablation

We further evaluate the effect of register-token conditioning in the latent world model on LIBERO (spatial suite). The variant without register-token conditioning removes the register-token input. As

shown in Figure 9, removing register-token conditioning decreases the success rate from 92.8% to 91.6%. This result suggests that register tokens provide useful global context beyond the target-view patch tokens in this setting.

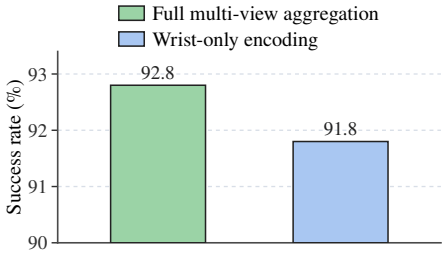

Figure 8: **Multi-view aggregation.** Ablation of multi-view geometry aggregation on LIBERO (spatial suite).

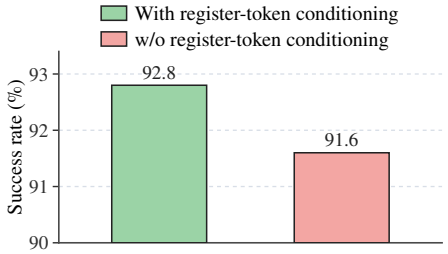

Figure 9: **Register-token conditioning.** Ablation of register-token conditioning on LIBERO (spatial suite).

C Real-World Evaluation Details

C.1 Hardware and Data Collection

We conduct the real-world evaluation on an SO-101 robot, a 6-DoF manipulation platform supported by the open-source HuggingFace LeRobot framework. The sensing setup contains one static camera that provides a third-person view of the workspace and one wrist-mounted camera that provides a first-person view near the gripper. This two-view setup matches the multi-view configuration used by GWM-VLA and exposes both global scene layout and local gripper-object interactions.

The SO-101 platform uses a Leader-Follower teleoperation design. During data collection, the operator physically moves the Leader arm, and the Follower arm synchronously executes the corresponding motion in the workspace. The robot states, actions, camera observations, and language instructions are recorded alongside each demonstration trajectory. We use this setup to collect 100 demonstrations over five pick-and-place training tasks, with 20 demonstrations per task.

C.2 Evaluation Protocol

We evaluate three real-world settings. The ID setting uses three seen pick-and-place instructions from the training task set: placing the green sauce packet on the brown cork plate, placing the black gear on the brown cork plate, and placing the orange sauce packet on the silver metal plate. The OOD-task setting evaluates held-out object–receptacle recombinations: placing the black gear on the silver metal plate, placing the green sauce packet on the brown box, and placing the orange sauce packet on the brown box. The OOD-layout setting uses the same three ID instructions but changes the object arrangements in the workspace. Each task is evaluated over 10 trials. Table 6 reports task-level success counts for the real-world SO-101 evaluation.

C.3 Analysis

The real-world results reveal different generalization patterns across the evaluation settings. On the ID tasks, π_0, $\pi_{0.5}$, and GWM-VLA all achieve strong success rates, indicating that the seen object–receptacle configurations are handled reliably by all three methods. Under OOD-layout shifts, $\pi_{0.5}$ and GWM-VLA retain substantially better performance than π_0, while GWM-VLA achieves a slightly higher average success rate than $\pi_{0.5}$. This robustness to rearranged object configurations suggests that the geometry-aware representation in GWM-VLA provides more effective spatial awareness under layout shifts.

On the OOD-task setting, GWM-VLA achieves a slightly lower aggregate success rate than $\pi_{0.5}$. In trials involving the previously unseen brown box, we observed occasional failures in which $\pi_{0.5}$ failed to lift the object high enough and collided with the box during transport; this failure pattern did not occur for GWM-VLA in our evaluation. One possible explanation is that geometry-aware

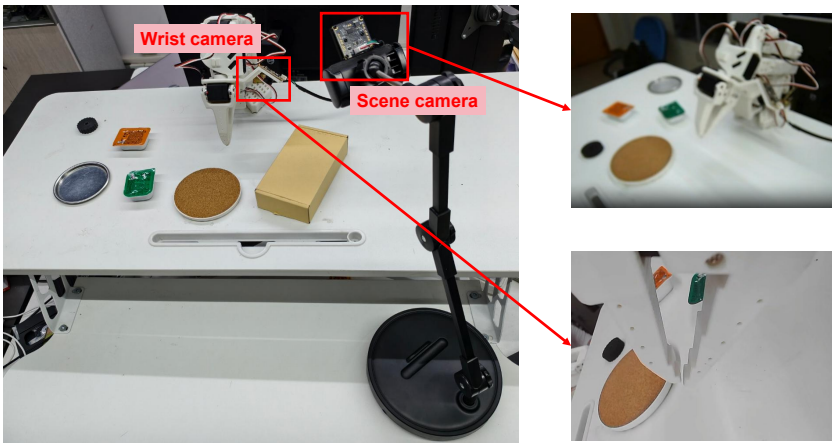

Figure 10: Setup of the real-world SO-101 robot evaluation experiments, consisting of one scene camera and one wrist camera.

Table 6: Real-world SO-101 success counts (successes/10 trials) with average success rates. The best and runner-up results are highlighted.

Setting	Task	π_0	$\pi_{0.5}$	GWM-VLA
ID	Pick up the green sauce packet and place it on the brown cork plate.	5/10	7/10	**8/10**
ID	Pick up the black gear and place it on the brown cork plate.	6/10	**8/10**	7/10
ID	Pick up the orange sauce packet and place it on the silver metal plate.	6/10	6/10	**9/10**
OOD Task	Pick up the black gear and place it on the silver metal plate.	1/10	**6/10**	5/10
OOD Task	Pick up the green sauce packet and place it on the brown box.	0/10	**6/10**	**6/10**
OOD Task	Pick up the orange sauce packet and place it on the brown box.	1/10	**6/10**	5/10
OOD Layout	Pick up the green sauce packet and place it on the brown cork plate.	1/10	**7/10**	**7/10**
OOD Layout	Pick up the black gear and place it on the brown cork plate.	3/10	**6/10**	**6/10**
OOD Layout	Pick up the orange sauce packet and place it on the silver metal plate.	1/10	5/10	**7/10**
Avg.	All evaluation tasks	26.7%	63.3%	**66.7%**

latent supervision helps the policy account for height-related spatial constraints. However, additional trajectory-level analysis is required to verify this hypothesis.