# WAVE-0: BRIDGING VLA POLICIES AND WAMS WITH LIGHTWEIGHT LATENT FUTURE PREDICTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Vision–language–action (VLA) policies excel at generalist robotic manipulation but generate actions without explicitly modeling their environmental consequences. World action models (WAMs) incorporate future prediction, yet most rely on costly video generation, while latent alternatives either use predicted futures only as auxiliary targets or infer latent actions from suboptimal visual transitions through costly large-scale video pretraining. We introduce **WAVE-0** (**W**orld-model-**A**ugmented **V**LA **E**xecution), a framework that bridges VLAs and WAMs through a lightweight latent world model. By connecting the semantic understanding of pretrained VLAs with explicit future prediction, WAVE-0 allows anticipated action consequences to directly guide action generation. Built on a pretrained VLA, its 257M world model translates VLM-generated latent action plans into predicted future representations that condition the action expert, unifying latent planning, future prediction, and action generation. The world model is trained directly on ground-truth robot actions with a counterfactual ranking objective, requiring neither pixel reconstruction nor large-scale video pretraining. WAVE-0 achieves success rates of 91.47% on RoboTwin 2.0 and 98.7% on LIBERO, along with 88.0%, 61.4%, and 75.0% on LIBERO-Plus, LIBERO-Pro, and LIBERO-Para, respectively. On real-world tasks, it improves the success rate of $\pi_{0.5}$ from 38.3% to 73.3%. These results demonstrate strong manipulation performance and generalization across diverse distribution shifts. Code will be released.

## 1 INTRODUCTION

Vision–language–action (VLA) policies have made substantial progress in general robotic manipulation by combining the semantic knowledge of pretrained vision–language models with action supervision from robot demonstrations (Kim et al., 2024; Black et al., 2026; Intelligence et al., 2025; Bjorck et al., 2025; Brohan et al., 2022; 2023; Liu et al., 2025). While effective, this direct observation-to-action mapping does not explicitly capture how actions transform the environment and thereby shape subsequent decisions. World action models (WAMs) address this limitation by incorporating environmental dynamics into action generation, enabling policies to anticipate the future consequences of their actions (Ye et al., 2026b; Bi et al., 2026; Li et al., 2026). Most existing WAMs build on large-scale video generation models, whose pixel-level prediction spends substantial computation on visual details irrelevant to action selection and limits inference efficiency. Recent efforts have tried to reduce this overhead (Yuan et al., 2026; Cai et al., 2026), but they still rely on video generation backbones that devote considerable capacity to visual reconstruction.

A promising alternative is to model environmental dynamics in a compact latent space, allowing future prediction to focus on information relevant to action selection rather than visual reconstruction. VLA-JEPA (Sun et al., 2026) learns latent actions through future-state prediction, while JEPA-WAM (Lin et al., 2026) uses a shared predictor to model state transitions and generate actions. However, neither approach explicitly conditions action generation on predicted future states, leaving future prediction and action selection only loosely connected. LaWAM (Chen et al., 2026) guides action generation with predicted latent visual subgoals, but learns its latent actions indirectly from visual transitions through inverse dynamics. While this formulation enables world model pretraining on large-scale videos without action annotations, the inferred latent actions may summarize realized transitions rather than the executable actions that caused them. Consequently, a representation that explains an observed transition may be misaligned with the action a policy must generate before



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.

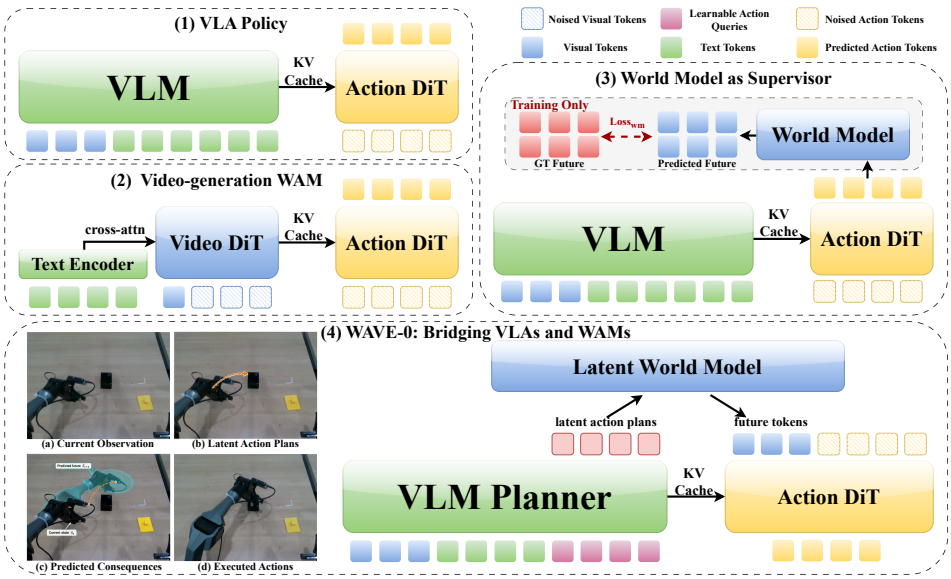



Figure 1: **Comparison of representative robot policy paradigms.** (1) Standard VLAs map vision-language context directly to actions. (2) Video-based WAMs model dynamics through costly video generation. (3) A world model provides auxiliary supervision by predicting the consequences of generated actions during training, leaving the inference pathway unchanged. (4) WAVE-0 translates VLM-generated latent action plans into predicted future representations with a lightweight latent world model, conditioning the action expert during both training and inference.

the transition occurs. Moreover, LaWAM pretrains its world model on approximately 4,500 hours of videos, followed by policy-integration pretraining and additional benchmark-specific training, resulting in substantial training costs. This raises a natural question: *when domain-specific demonstrations already provide ground-truth executable actions, is such large-scale, indirect pretraining truly more effective than directly training the world model with those actions?*

We answer this question by grounding future prediction directly in the ground-truth actions of robot trajectories. Our key insight is that demonstrations already pair each action sequence with its resulting observations, providing direct supervision on how actions change the environment. A world model trained with this supervision predicts futures tied to the executed actions rather than to latent actions inferred after the transition occurs, and requires no large-scale pretraining. Moreover, since a pretrained VLA already provides rich semantic understanding and a visual representation space, the world model only needs to predict how this representation evolves under a given action plan, which keeps it lightweight and avoids reconstructing future images. We therefore view semantic understanding in VLAs and future prediction in WAMs as complementary, where the former determines action intent and the latter anticipates its consequences. Building on this view, we introduce **WAVE-0** (**W**orld-model-**A**ugmented **V**LA **E**xecution), a framework that bridges VLAs and WAMs through action-grounded latent future prediction. As shown in Figure 1, our latent world model transforms VLM-generated latent action plans into predicted future visual representations that explicitly condition the action expert, yielding a unified pipeline of *latent planning, future prediction, and action generation*.

WAVE-0 follows a two-stage training strategy. In the first stage, we train the latent world model on action-labeled trajectories to predict future visual representations from current observations and ground-truth actions. Besides future and latent delta objectives, a counterfactual ranking objective encourages the model to distinguish the consequences of different action plans. In the second stage, we integrate the frozen world model into pretrained $\pi_{0.5}$ (Intelligence et al., 2025) and align VLM-generated latent plans with the learned action representation space. The world model predicts future representations from these plans, and a lightweight resampler compresses them into future-conditioning tokens for the action expert. The planning branch guides action generation only through these predictions, while the expert retains access to the original VLM context. This design

allows anticipated action consequences to explicitly guide action generation during both training and inference.

On RoboTwin 2.0 (Chen et al., 2025) and LIBERO (Liu et al., 2023), WAVE-0 achieves average success rates of 91.47% and 98.7%, improving over $\pi_{0.5}$ by 11.09 and 1.8 percentage points, respectively. On the more challenging LIBERO-Plus (Fei et al., 2025) and LIBERO-Pro (Zhou et al., 2025), which introduce variations in objects, layouts, camera viewpoints, and instructions, WAVE-0 achieves 88.0% and 61.4%, outperforming all compared methods. On LIBERO-Para (Kim et al., 2026a), it exceeds $\pi_{0.5}$ by 3.6 percentage points under instruction paraphrasing. These gains extend to three real-world tasks, where WAVE-0 improves the average success rate of $\pi_{0.5}$ from 38.3% to 73.3%. Compared with LaWAM, WAVE-0 is also more training-efficient, processing far fewer training examples across RoboTwin 2.0 and real-world tasks. Together, these results demonstrate that action-grounded latent future prediction not only improves a strong VLA with modest cost, but also substantially enhances its robustness to visual, spatial, and language distribution shifts.

Our contributions are threefold:

(1) We introduce WAVE-0, a framework that connects VLA semantic understanding with WAM future prediction through a 257M lightweight latent world model, enabling anticipated action consequences to guide action generation during training and inference.

(2) We develop a two-stage training strategy that learns action-dependent latent dynamics from ground-truth robot actions using counterfactual ranking, then integrates the frozen world model into a pretrained VLA through an explicit prediction bottleneck.

(3) Evaluations on RoboTwin 2.0, LIBERO, and its generalization benchmarks demonstrate WAVE-0's effectiveness and generalization under visual and language distribution shifts. Real-world experiments further demonstrate substantial improvements over $\pi_{0.5}$ across three manipulation tasks.

## 2 Related Works

### 2.1 Vision-Language-Action Policies

Vision-language-action (VLA) policies have advanced generalist robotic manipulation by combining pretrained vision-language knowledge with robot action supervision (Brohan et al., 2022; 2023; Kim et al., 2024; 2025; Team et al., 2025; Yang et al., 2026b). Diffusion-based and flow-based policies further support expressive continuous action generation across diverse tasks and embodiments (Black et al., 2026; Intelligence et al., 2025; Bjorck et al., 2025; Liu et al., 2025; Wen et al., 2025). Beyond action generation, embodied reasoning and explicit spatial representations have also been explored to enhance VLA policies (Zawalski et al., 2024; Qu et al., 2025). Another line of work learns task-centric latent action representations from videos, enabling VLA pretraining across embodiments with reduced reliance on action annotations (Ye et al., 2025; Bu et al., 2025; Yang et al., 2026a). Despite these advances, VLAs primarily map observations and instructions to actions, leaving action-dependent environmental dynamics implicit. This motivates integrating explicit future prediction into VLAs, allowing anticipated action consequences to directly guide action generation.

### 2.2 Visual Generation-Based World Action Models

World action models (WAMs) incorporate future-state prediction into action generation to explicitly model how robot actions affect the environment. Most existing WAMs build on pretrained video models that jointly predict future observations and actions (Ye et al., 2026b; Bi et al., 2026; Kim et al., 2026b; Li et al., 2026; Pai et al., 2025; Liang et al., 2025; Liao et al., 2026). However, large backbones, modeling of action-irrelevant visual details, and iterative denoising incur substantial computational cost and latency. To reduce this cost, Fast-WAM (Yuan et al., 2026) and GigaWorld-Policy (Ye et al., 2026a) retain visual prediction only as training supervision and skip future generation at inference, while other methods reuse future information through asynchronous execution, sparse interactions, or single-pass caching (Cai et al., 2026; Zhao et al., 2026; Zhang et al., 2026a). Beyond video generation, BagelVLA (Hu et al., 2026) builds on a unified model for understanding and generation to extract predictive visual features through single-step denoising, while ImageWAM (Zhang et al., 2026b) guides action generation using denoising features from an image-

editing model without decoding future images at inference. However, visual generation backbones remain computationally demanding, motivating lightweight dynamics models that directly predict future visual representations without pixel-level reconstruction.

### 2.3 Latent World Action Models

Joint-embedding predictive architecture (JEPA) offers a framework for latent world action modeling by predicting representations without pixel reconstruction (LeCun et al., 2022; Assran et al., 2023; 2025; Maes et al., 2026). VLA-JEPA (Sun et al., 2026) learns action-relevant latent representations from action-free human videos through world prediction, and then adapts them to robot control. JEPA-WAM (Lin et al., 2026) uses spatially structured targets that jointly encode current and future observations, providing dense transition supervision for policy learning. However, both methods use predictive supervision to shape policy representations without directly conditioning action generation on predicted visual representations. LaWAM (Chen et al., 2026) learns latent actions through visual inverse dynamics and uses them to predict visual subgoals that guide action generation. We posit that semantic understanding in pretrained VLAs and future prediction in WAMs are complementary capabilities that should jointly guide action generation. WAVE-0 bridges these paradigms through a lightweight latent world model that learns to predict world state transitions from labeled robot action sequences. It translates VLM-generated latent action plans into future representations that explicitly guide the action expert during both training and inference.

## 3 Method

### 3.1 Problem Formulation

We consider robot manipulation conditioned on a current observation and a task instruction. At time $t$, the robot receives an observation $o_t = (I_t, s_t)$, where $I_t = \{I_t^{(c)}\}_{c=1}^{C}$ contains images from $C$ cameras and $s_t$ is the proprioceptive state. Given an instruction $\ell$, the robot generates a continuous action chunk with an action horizon of $H$:

$$A_t = (a_t, \ldots, a_{t+H-1}), \qquad a_t \in \mathbb{R}^{d_a}. \tag{1}$$

We learn a policy from demonstrations to generate actions conditioned on the current observation and instruction:

$$A_t \sim \pi_\theta(\cdot \mid o_t, \ell). \tag{2}$$

WAVE-0 decomposes this process into latent action planning, future world state prediction, and future-conditioned action generation. Let $S_t = \mathcal{E}(I_t)$ denote the world state representation in a fixed visual embedding space. A VLM planner $f_\eta$ produces a latent action plan, a world model $g_\phi$ predicts its consequences, and an action expert $p_\psi$ generates actions using both the original context and the predicted future:

$$\widehat{Z}_a = f_\eta(o_t, \ell), \quad \widehat{S}_{t+h} = g_\phi(S_t, s_t, \widehat{Z}_a), \quad A_t \sim p_\psi(\cdot \mid o_t, \ell, \widehat{S}_{t+h}), \tag{3}$$

where $h$ denotes the future observation offset.

### 3.2 Model Architecture

Built on pretrained $\pi_{0.5}$ (Intelligence et al., 2025), WAVE-0 combines vision-language understanding with explicit dynamics prediction through latent action planning, future world state prediction, and future-conditioned action generation. The training procedure is conducted in two sequential stages. In the first stage, the latent world model jointly learns action representations and latent dynamics from visual observations, proprioceptive states, and ground-truth action sequences to predict future world states in a fixed visual embedding space. In the second stage, we integrate the frozen world model into $\pi_{0.5}$ and train the VLM to produce latent action plans aligned with the learned action representation space. The world model converts these plans into predicted future visual representations, which condition the action expert alongside the original vision-language context.

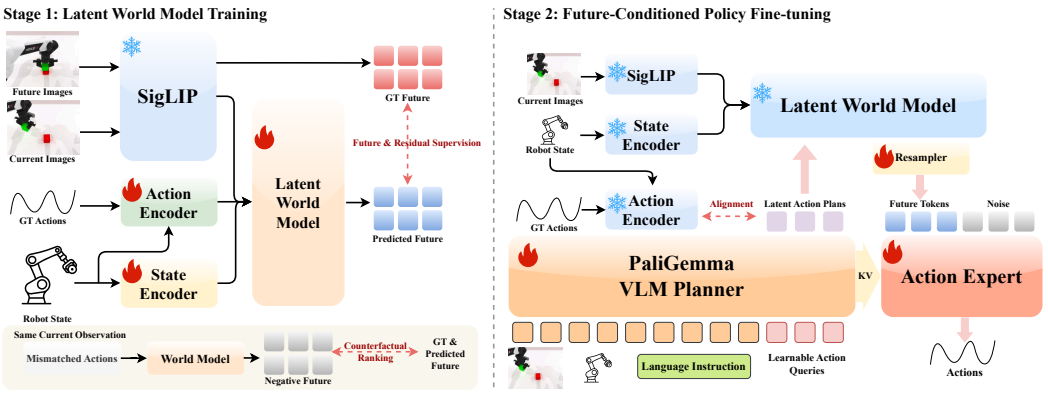



Figure 2: **Overview of WAVE-0. Stage 1:** The latent world model learns action-conditioned dynamics in a fixed visual embedding space, trained with future-representation prediction, latent residual supervision, and counterfactual action ranking. **Stage 2:** The VLM generates latent action plans aligned with the representations from the frozen action encoder. The frozen world model predicts the visual consequences of these plans, and a resampler compresses them into future tokens that condition the action expert alongside the original VLM context.

**Latent World Model.** We learn action-conditioned latent dynamics in a fixed visual embedding space. We obtain $S_t = \mathcal{E}(I_t)$ and $S_{t+h} = \mathcal{E}(I_{t+h})$ using the frozen SigLIP (Zhai et al., 2023) encoder and visual projector from pretrained $\pi_{0.5}$, followed by spatial pooling. An action encoder embeds the ground-truth action sequence conditioned on the current proprioceptive state, yielding $Z_a = \mathcal{E}_a(A_t, s_t)$, while a state encoder produces $z_s = \mathcal{E}_s(s_t)$. A Transformer with action-conditioned AdaLN-Zero blocks (Peebles & Xie, 2023) predicts a latent residual, which is added to the current representation to obtain the predicted future:

$$\Delta\widehat{S} = \mathcal{W}_\phi(S_t, z_s, Z_a), \quad \widehat{S}_{t+h} = S_t + \Delta\widehat{S}. \tag{4}$$

We supervise the future representation with a cosine distance loss and the state change with a SmoothL1 loss:

$$\mathcal{L}_{\text{future}} = d_{\cos}(\widehat{S}_{t+h}, S_{t+h}), \quad \mathcal{L}_{\text{delta}} = \text{SmoothL1}(\Delta\widehat{S}, S_{t+h} - S_t), \tag{5}$$

where $d_{\cos}$ denotes mean token-wise cosine distance.

To encourage action dependence, we introduce a counterfactual ranking loss. Keeping the current observation fixed, we replace the demonstrated actions with a mismatched sequence $\widetilde{A}_t$ to obtain:

$$\widehat{S}^-_{t+h} = S_t + \mathcal{W}_\phi(S_t, z_s, \mathcal{E}_a(\widetilde{A}_t, s_t)). \tag{6}$$

The loss favors demonstrated actions over mismatched actions when predicting the same observed future:

$$\mathcal{L}_{\text{rank}} = r\left[m + d_{\cos}(\widehat{S}_{t+h}, S_{t+h}) - d_{\cos}(\widehat{S}^-_{t+h}, S_{t+h})\right]_+, \tag{7}$$

where $[x]_+ = \max(0, x)$, $m = 0.1$, and $r \in \{0, 1\}$ masks unchanged negative sequences.

We jointly optimize the action encoder, state encoder, and world model using the following objective:

$$\mathcal{L}_{\text{stage1}} = \mathcal{L}_{\text{future}} + 0.5\,\mathcal{L}_{\text{delta}} + 0.1\,\mathcal{L}_{\text{rank}}. \tag{8}$$

**Latent Action Planning.** We integrate the frozen latent world model into pretrained $\pi_{0.5}$ and train its PaliGemma VLM (Beyer et al., 2024) to generate latent action plans in the action representation space learned in Stage 1. Specifically, we append learnable latent action queries $Q$ to the VLM input sequence and transform the output hidden states $U_Q$ into latent action tokens via $\widehat{Z}_a = \Pi_a(\text{LN}(U_Q))$, where $\Pi_a$ is a learned linear projection. These tokens condition the frozen world model to predict future visual representations:

$$\widehat{S}_{t+h} = S_t + \mathcal{W}_\phi(S_t, z_s, \widehat{Z}_a). \tag{9}$$

We supervise latent action planning by aligning $\widehat{Z}_a$ with the target representations $Z_a$ produced by the frozen action encoder:

$$\mathcal{L}_{\text{align}} = d_{\cos}(\widehat{Z}_a, Z_a) + 0.5\ \text{SmoothL1}(\widehat{Z}_a, Z_a). \tag{10}$$

To supervise the predicted visual consequences, we reuse the future representation and residual losses in Eq. 5, evaluated with $\widehat{Z}_a$ as the action condition:

$$\mathcal{L}_{\text{world}} = \mathcal{L}_{\text{future}} + 0.5\,\mathcal{L}_{\text{delta}}. \tag{11}$$

**Future-Conditioned Action Generation.** A lightweight resampler pools and projects the predicted future representation into conditioning tokens $F_t = \mathcal{R}(\widehat{S}_{t+h})$. We prepend these tokens to the noisy action tokens in the action expert, enabling it to attend to both the original VLM context and the predicted future. We exclude latent-action query tokens from the context passed to the expert, so that the planning branch guides action generation only through its predicted visual consequences.

In Stage 2, we jointly optimize the VLM planner, future-conditioning resampler, and action expert while keeping the world model frozen:

$$\mathcal{L}_{\text{stage2}} = \mathcal{L}_{\text{flow}} + \lambda_{\text{WAM}}\left(\mathcal{L}_{\text{world}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}}\right), \tag{12}$$

where $\mathcal{L}_{\text{flow}}$ is the flow-matching objective used in $\pi_{0.5}$, $\lambda_{\text{WAM}}$ scales the auxiliary supervision, and $\lambda_{\text{align}}$ controls the relative weight of latent-action alignment.

# 4 Experiments

## 4.1 Implementation Details

We initialize the VLM and action expert from pretrained $\pi_{0.5}$. For world modeling, we use a frozen copy of its SigLIP encoder and visual projector to extract 2048-dimensional visual features, which are spatially pooled into a $4 \times 4$ grid, yielding 16 tokens per camera. The latent world model processes these representations using a 12-layer Transformer with a hidden dimension of 1024, 16 attention heads, and an MLP dimension of 4096. The resampler then pools the predicted future representations to a $2 \times 2$ grid per camera and projects them into 12 conditioning tokens for the action expert, which retains the original flow-matching formulation of $\pi_{0.5}$.

The world model, action encoder, and state encoder together comprise approximately 257M trainable parameters in Stage 1, while the full WAVE-0 contains approximately 4.3B parameters. We train with AdamW using a default peak learning rate of $10^{-4}$, weight decay of 0.01, and a cosine learning rate schedule. Benchmark-specific training and inference configurations are provided in Appendix A.1.

Table 1: **Success rates (%) on LIBERO and RoboTwin 2.0.** Bold and underlined values indicate the best and second-best results, respectively. $\Delta$ denotes improvement over $\pi_{0.5}$.

(a) Results on LIBERO.

| Method | Spatial | Object | Goal | Long | Avg. |
|---|---|---|---|---|---|
| OpenVLA | 84.7 | 88.4 | 79.2 | 53.7 | 76.5 |
| GR00T N1 | 94.4 | 97.6 | 93.0 | 90.6 | 93.9 |
| $\pi_0$ | 96.8 | 98.8 | 95.8 | 85.2 | 94.1 |
| Motus | 96.8 | <u>99.8</u> | 96.6 | 97.6 | 97.7 |
| LingBot-VA | 98.5 | 99.6 | 97.2 | **98.5** | 98.5 |
| Fast-WAM | 98.2 | **100.0** | 97.0 | 95.2 | 97.6 |
| ImageWAM | 97.2 | 99.2 | **98.8** | <u>98.4</u> | 98.4 |
| VLA-JEPA | 96.2 | 99.6 | 97.2 | 95.8 | 97.2 |
| JEPA-WAM | 95.6 | 99.4 | 97.2 | 94.6 | 96.7 |
| LaWAM | **99.4** | 99.6 | 98.4 | 97.0 | <u>98.6</u> |
| $\pi_{0.5}$ | <u>98.8</u> | 98.2 | 98.0 | 92.4 | 96.9 |
| **WAVE-0 (Ours)** | 98.6 | **100.0** | <u>98.6</u> | 97.6 | **98.7** |
| $\Delta$ *Improvement* | *-0.2* | *+1.8* | *+0.6* | *+5.2* | *+1.8* |

(b) Results on RoboTwin 2.0.

| Method | Clean | Randomized | Avg. |
|---|---|---|---|
| $\pi_0$ | 65.92 | 58.40 | 62.16 |
| ABot-M0 | 81.20 | 80.40 | 80.80 |
| Motus | 88.66 | 87.02 | 87.84 |
| GigaWorld-Policy | 86.36 | 85.04 | 85.70 |
| LingBot-VA | 92.90 | 91.50 | 92.20 |
| Fast-WAM | 91.88 | 91.78 | 91.83 |
| AHA-WAM | **93.40** | <u>92.20</u> | <u>92.80</u> |
| LaWAM | 92.64 | 89.80 | 91.22 |
| ImageWAM | <u>93.20</u> | **93.56** | **93.38** |
| $\pi_{0.5}$ | 81.12 | 79.64 | 80.38 |
| $\pi_{0.5}$ + WM | 89.48 | 89.54 | 89.51 |
| **WAVE-0 (Ours)** | 91.46 | 91.48 | 91.47 |
| $\Delta$ *Improvement* | *+10.34* | *+11.84* | *+11.09* |

### 4.2 EVALUATION ON LIBERO

We evaluate on the four LIBERO suites: Spatial, Object, Goal, and Long, covering 40 tasks. Our training data comprise 1,693 trajectories and 273,465 frames across these suites. As shown in Table 1a, WAVE-0 achieves an average success rate of 98.7%, the highest among the compared methods, improving over $\pi_{0.5}$ by 1.8 percentage points. The largest gain occurs on the Long suite, where success increases from 92.4% to 97.6%. This suggests that anticipating action consequences is particularly beneficial for sequential manipulation, where the outcome of each action affects subsequent steps. WAVE-0 also achieves 100.0% on Object and 98.6% on Goal, supporting the effectiveness of latent future conditioning across diverse manipulation tasks.

### 4.3 EVALUATION ON ROBOTWIN 2.0

RoboTwin 2.0 is a bimanual manipulation benchmark with 50 tasks. Following prior work (Bi et al., 2026; Li et al., 2026; Yuan et al., 2026), we train a single policy on all tasks using 2,500 demonstrations collected in clean scenes and 25,000 demonstrations collected under heavy domain randomization.

As shown in Table 1b, WAVE-0 achieves an average success rate of 91.47%, improving over $\pi_{0.5}$ by 11.09 percentage points, with consistent gains in both clean (81.12% to 91.46%) and randomized (79.64% to 91.48%) scenes. Compared with LaWAM, which also models dynamics in latent space, WAVE-0 achieves a higher average success rate (91.47% vs. 91.22%) and remains more robust under randomization (91.48% vs. 89.80%). Per-task results in Appendix Table 13 further show large gains over $\pi_{0.5}$ on long-horizon tasks with multiple dependent subgoals, such as Stack Blocks Three (+23.0) and Stack Bowls Three (+33.0), suggesting that anticipating the consequences of latent action plans benefits sequential manipulation.

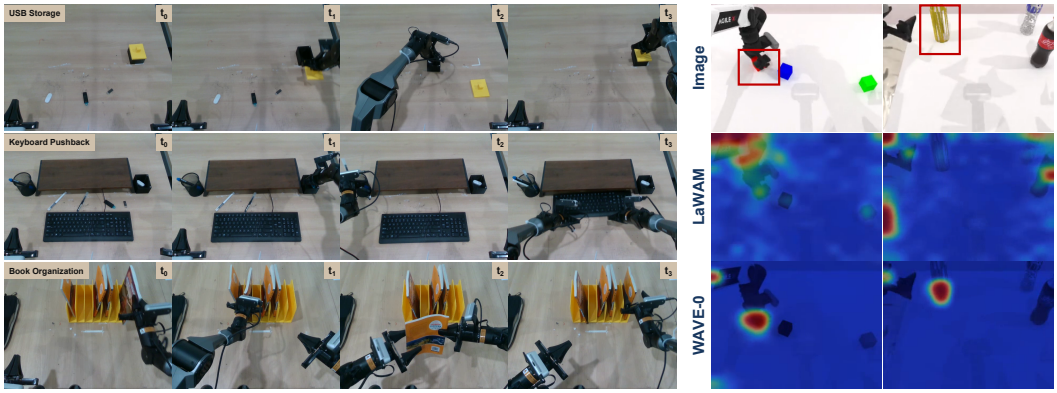
Table 2: **Success rates (%) on LIBERO-Plus and LIBERO-Pro.** Bold and underlined values indicate the best and second-best results, respectively. $\Delta$ denotes improvement over $\pi_{0.5}$.

| | LIBERO-Plus | | | | | | | | LIBERO-Pro |
|---|---|---|---|---|---|---|---|---|---|
| **Method** | **Camera** | **Robot** | **Language** | **Light** | **Background** | **Noise** | **Layout** | **Avg.** | **Avg.** |
| OpenVLA | 0.8 | 3.5 | 23.0 | 8.1 | 34.8 | 15.2 | 28.5 | 15.6 | 52.0 |
| Fast-WAM | 16.4 | 44.5 | 68.9 | 78.2 | 53.7 | 37.7 | 60.7 | 51.5 | 40.2 |
| ImageWAM | **80.8** | 50.3 | **91.4** | <u>98.1</u> | 85.5 | **93.8** | 80.5 | 83.1 | 48.7 |
| VLA-JEPA | 63.3 | 67.1 | 85.4 | 95.6 | 93.6 | 66.3 | 85.1 | 79.5 | 48.3 |
| JEPA-WAM | <u>79.2</u> | 59.2 | 68.2 | 93.3 | 94.6 | 83.6 | 76.1 | 79.2 | 41.9 |
| LaWAM | 37.2 | 71.6 | <u>87.1</u> | 93.5 | 94.1 | 69.4 | 83.3 | 76.6 | 52.7 |
| $\pi_{0.5}$ | 69.4 | <u>75.3</u> | 82.6 | 96.7 | <u>96.8</u> | <u>84.3</u> | <u>86.2</u> | <u>84.5</u> | <u>53.0</u> |
| **WAVE-0 (Ours)** | 75.9 | **89.3** | 86.1 | **98.4** | **98.1** | 80.6 | **87.7** | **88.0** | **61.4** |
| $\Delta$ *Improvement* | *+6.5* | *+14.0* | *+3.5* | *+1.7* | *+1.3* | *-3.7* | *+1.5* | *+3.5* | *+8.4* |

### 4.4 OUT-OF-DISTRIBUTION GENERALIZATION

We evaluate generalization on LIBERO-Plus, LIBERO-Pro, and LIBERO-Para, directly using the model trained on LIBERO without additional fine-tuning. LIBERO-Plus evaluates robustness across seven perturbation categories, including camera viewpoints, robot initial states, language instructions, lighting, backgrounds, sensor noise, and object layouts. As shown in Table 2, WAVE-0 achieves an overall success rate of 88.0%, outperforming all compared methods and exceeding $\pi_{0.5}$ by 3.5 percentage points. It improves performance in six of the seven categories, with the largest gains under robot initial-state perturbations (+14.0) and camera viewpoint changes (+6.5).

On LIBERO-Pro, which measures generalization under changes to objects, spatial configurations, instructions, and environments, WAVE-0 achieves 61.4% success, the highest among the compared methods and an 8.4-point improvement over $\pi_{0.5}$. We further assess robustness to instruction paraphrasing on LIBERO-Para, where WAVE-0 improves success from 71.4% to 75.0%. Together, these results indicate that latent future conditioning improves generalization under visual, spatial, and linguistic distribution shifts.




*(a) Real-world robot rollouts* *(b) Attention visualization*

Figure 3: **Real-world rollouts and attention visualization.** (a) Key frames from WAVE-0 rollouts on the three real-world tasks. (b) Attention maps on two RoboTwin 2.0 tasks.

Table 3: **Real-world success rates (%).** Each task is evaluated over 20 trials.

| Method | Book | USB | Keyboard | Avg. |
|---|---|---|---|---|
| ImageWAM | 50.0 | 25.0 | 45.0 | 40.0 |
| LaWAM | 40.0 | 35.0 | 55.0 | 43.3 |
| $\pi_{0.5}$ | 40.0 | 45.0 | 30.0 | 38.3 |
| **WAVE-0** | **85.0** | **65.0** | **70.0** | **73.3** |
| $\Delta$ *Improvement* | *+45.0* | *+20.0* | *+40.0* | *+35.0* |

Table 4: **Comparison of training sample efficiency.**

| Training stage | LaWAM | WAVE-0 |
|---|---|---|
| Shared pretraining | 307.20M | 10.24M |
| Policy per task | 12.80M | 7.68M |
| Total for three tasks | 345.60M | 33.28M |
| **Amortized per task** | **115.20M** | **11.09M** |

### 4.5 Evaluation on Real-World Tasks

We evaluate WAVE-0 on three real-world tasks with a dual-arm AgileX Piper platform, namely Book Organization, USB Storage, and Keyboard Pushback, which require rearranging objects in clutter, precise insertion, and long-horizon bimanual coordination, respectively. For each task, we collect 200 demonstrations, train a separate model for each method on the same data, and evaluate it over 20 trials with diverse initial configurations (Figure 3(a)). As shown in Table 3, WAVE-0 achieves the best performance on all three tasks, reaching an average success rate of 73.3% compared with 43.3% for the strongest baseline LaWAM and 40.0% for ImageWAM. Relative to $\pi_{0.5}$, WAVE-0 nearly doubles the average success rate from 38.3% to 73.3%.

### 4.6 Analysis

**Training Efficiency.** WAVE-0 learns latent dynamics from action-labeled target demonstrations without large-scale video pretraining. In contrast, LaWAM performs 100k steps of world model pretraining and 200k steps of policy-integration pretraining with a global batch size of 1,024, processing 307.2M examples before benchmark-specific training. On RoboTwin 2.0, the two-stage training of WAVE-0 requires approximately 670 H100 GPU hours and processes 76.8M examples, compared with 1,280 H100 GPU hours and 102.4M examples for LaWAM's benchmark-specific training alone. Across three real-world tasks, WAVE-0 processes 33.28M examples, compared with 345.60M for LaWAM, including shared pretraining and task-specific policy training. Amortized per task, these budgets are 11.09M and 115.20M, respectively, with WAVE-0 achieving higher success rates, as shown in Tables 4 and 3. These results support the efficiency of action-grounded latent world modeling without additional large-scale video pretraining.

**Attention Visualization.** Figure 3(b) compares attention from action queries to current main-view image patches at the final denoising step. In the illustrated examples, WAVE-0 exhibits more localized attention around the manipulated object and target bottle, whereas LaWAM shows more dispersed responses over the robot arm, image boundaries, and background regions.

### 4.7 Ablation Studies

#### 4.7.1 Effects of World Model Integration Strategy

We investigate how the learned world model should be integrated into the policy by comparing $\pi_{0.5}$, $\pi_{0.5}$ + WM, and WAVE-0 on RoboTwin 2.0. As illustrated in Figure 1, $\pi_{0.5}$ + WM uses the world model to predict future representations from generated actions and provides an auxiliary training loss against the observed future. In contrast, WAVE-0 explicitly conditions action generation on predicted futures during both training and inference. Both variants use the same frozen world model trained in Stage 1.

As shown in Table 1b, auxiliary supervision improves average success from 80.38% to 89.51%, supporting the value of the learned latent dynamics for policy learning. WAVE-0 further increases success to 91.47%, outperforming $\pi_{0.5}$ + WM by 1.98 and 1.94 percentage points in clean and randomized settings, respectively. These results indicate that while world model supervision already yields substantial gains, explicitly conditioning action generation on predicted futures provides consistent additional benefits. More broadly, they support the central premise of WAM that anticipating the consequences of actions improves action generation, and show that this benefit can be realized in a compact latent space without video generation.

#### 4.7.2 Effects of Training Objectives

**Counterfactual ranking in Stage 1.** We remove $\mathcal{L}_{\text{rank}}$ while retaining $\mathcal{L}_{\text{future}}$ and $\mathcal{L}_{\text{delta}}$, and evaluate both variants on the same 2,048 samples from 256 episodes. Holding observations and robot states fixed, we measure the cosine prediction errors under demonstrated and mismatched actions against the same ground-truth future, denoted by $d_{\text{correct}}$ and $d_{\text{wrong}}$. We report the mean gap $g = d_{\text{wrong}} - d_{\text{correct}}$ over valid samples and the percentage satisfying $g \geq 0.1$, termed the action sensitivity gap and margin satisfaction rate, respectively. As shown in Figure 4(a,b), ranking supervision increases the gap to 0.1648 and margin satisfaction to 89.5% at 20k steps, compared with 0.0020 and 0% without it. It also improves success by 2.6 and 3.0 percentage points on LIBERO-Plus and LIBERO-Pro, respectively, supporting the value of action-dependent dynamics for generalization, as shown in Figure 4(c).

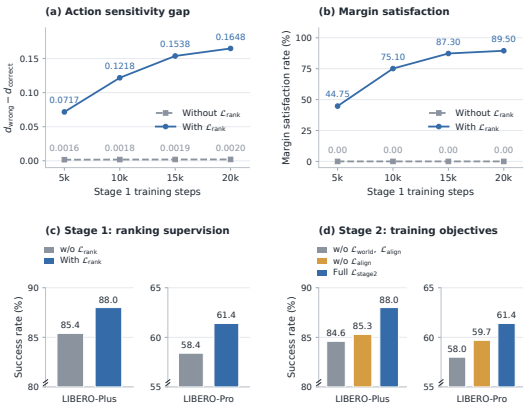



Figure 4: **Effects of training objectives.**

**Auxiliary objectives in Stage 2.** Starting from $\mathcal{L}_{\text{flow}}$ alone, adding $\mathcal{L}_{\text{world}}$ improves success by 0.7 and 1.7 percentage points on LIBERO-Plus and LIBERO-Pro, respectively. Adding $\mathcal{L}_{\text{align}}$ yields further gains of 2.7 and 1.7 points, reaching 88.0% and 61.4%, as shown in Figure 4(d). These results support the complementary benefits of supervising latent plans through predicted visual consequences and alignment with learned action representations.

## 5 Conclusion

We presented WAVE-0, a framework that bridges VLA policies and world action models through a lightweight latent world model. Instead of generating future frames or using predicted futures only as auxiliary training targets, WAVE-0 translates VLM-generated latent action plans into predicted future visual representations that explicitly condition the action expert during both training and inference. The world model is grounded in ground-truth robot actions and trained with counterfactual ranking supervision, which encourages action-dependent dynamics and improves policy generalization. Built on $\pi_{0.5}$, WAVE-0 substantially improves performance on simulation and real-world benchmarks without large-scale world model pretraining, suggesting that grounding latent dynamics in executable actions offers an efficient path toward world-model-augmented robot policies.

### AI USE STATEMENT

In this work, we used generative AI tools to assist with drafting and polishing manuscript text, identifying relevant literature, and code development. We reviewed and revised the AI-assisted content, checked references against the original sources, and verified that descriptions and conclusions were consistent with the methods and experimental results. We take responsibility for the final content of this work, including text, code, claims, and artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

We provide detailed experimental settings to facilitate reproduction of the results presented in this paper. Benchmark-specific training and evaluation configurations are available in Appendix A.1 and A.2.

## REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

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.

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

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026.

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

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. $\pi_0$: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

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.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

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*, 2025.

Jisong Cai, Long Ling, Shiwei Chu, Zhongshan Liu, Jiayue Kang, Zhixuan Liang, Wenjie Xu, Yinan Mao, Weinan Zhang, Xiaokang Yang, et al. Aha-wam: Asynchronous horizon-adaptive world-action modeling with observation-guided context routing. *arXiv preprint arXiv:2606.09811*, 2026.

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, et al. Lawam: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

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.

Yucheng Hu, Jianke Zhang, Yuanfei Luo, Yanjiang Guo, Xiaoyu Chen, Xinshu Sun, Kun Feng, Qingzhou Lu, Sheng Chen, Yangang Zhang, et al. Bagelvla: Enhancing long-horizon manipulation via interleaved vision-language-action generation. *arXiv preprint arXiv:2602.09849*, 2026.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL `https://arxiv.org/abs/2504.16054`.

Chanyoung Kim, Minwoo Kim, Minseok Kang, Hyunwoo Kim, and Dahuin Jung. Libero-para: A diagnostic benchmark and metrics for paraphrase robustness in vla models. *arXiv preprint arXiv:2603.28301*, 2026a.

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

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

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026b.

Yann LeCun et al. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Si Liu, Jianlan Luo, Liliang Chen, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 88446–88463, 2026.

Yihan Lin, Jiawei He, Shifeng Bao, Chen Zhao, Yang Li, Xiaobo Wang, Yan Wang, Cheng Chi, and Jing Zhang. Jepa-wam: Learning vision-language-action policies with joint-embedding world modeling. *arXiv preprint arXiv:2608.09381*, 2026.

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.

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

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

Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond vlas. *arXiv preprint arXiv:2512.15692*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

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. In *European Conference on Computer Vision*, pp. 478–497. Springer, 2026.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 2025.

Jiange Yang, Yansong Shi, Haoyi Zhu, Mingyu Liu, Kaijing Ma, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Como: Learning continuous latent motion from internet videos for scalable robot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42352–42363, 2026a.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026b.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. Gigaworld-policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026a.

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*, 2026b.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952. IEEE, 2023.

Chushan Zhang, Jinguang Tong, Xuesong Li, Yikai Wang, and Hongdong Li. Keep the future, drop the rollout: Rift for world action models. *arXiv preprint arXiv:2608.11521*, 2026a.

Yuyang Zhang, Wenyao Zhang, Zekun Qi, He Zhang, Haitao Lin, Jingbo Zhang, Yao Mu, Xiaokang Yang, Wenjun Zeng, and Xin Jin. Imagewam: Do world action models really need video generation, or just image editing? *arXiv preprint arXiv:2606.19531*, 2026b.

Weiheng Zhao, Haoyi Jiang, Xin Shi, Liu Liu, Fan Huang, Zhizhong Su, Wei Sui, and Xinggang Wang. Faster-wam: Efficient inference-time future conditioning for robust world action models. *arXiv preprint arXiv:2608.04404*, 2026.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025.

# A Appendix

## A.1 Training and Evaluation Details on Simulated Benchmarks

**LIBERO** All models are trained on 8 NVIDIA RTX PRO 6000 Blackwell GPUs with 96 GB memory per GPU, using Accelerate DDP, bf16 mixed precision, and the AdamW optimizer. We use a global batch size of 512, a peak learning rate of $1 \times 10^{-4}$, weight decay of $1 \times 10^{-2}$, and gradient clipping of 1.0. The learning rate is linearly warmed up for 1,000 steps followed by cosine decay. Training consists of two stages: the latent world model is pretrained for 20k steps, followed by 10k steps of action-policy training with the pretrained world model frozen. Stage 1 takes approximately 10 hours, while Stage 2 takes approximately 13 hours. For LIBERO, we use two camera views, a 7-dimensional action space, and an 8-dimensional proprioceptive state. The policy predicts 32-step action chunks and replans every 10 steps.

Table 5: **Training details of WAVE-0 on LIBERO.**

| Configuration | Stage 1: World Model | Stage 2: Action Policy |
|---|---|---|
| Number of GPUs | 8 | 8 |
| Precision | bf16 | bf16 |
| Optimizer | AdamW | AdamW |
| Global batch size | 512 | 512 |
| Peak learning rate | $1 \times 10^{-4}$ | $1 \times 10^{-4}$ |
| Final learning rate | $2.5 \times 10^{-6}$ | $1 \times 10^{-6}$ |
| Weight decay | $1 \times 10^{-2}$ | $1 \times 10^{-2}$ |
| Warmup steps | 1,000 | 1,000 |
| Gradient clipping | 1.0 | 1.0 |
| Training steps | 20,000 | 10,000 |
| Action chunk size $H$ | 32 | 32 |
| World model horizon $h$ | 9 | 9 |
| Latent action queries | – | 10 |
| World model | Trainable | Frozen |

**RoboTwin 2.0.** We train WAVE-0 on 8 NVIDIA H100 GPUs using Accelerate DDP and bf16 mixed precision, with a global batch size of 512. Stage 1 takes approximately 6 hours for 50k steps, while Stage 2 takes approximately 78 hours for 100k steps, with the world model frozen. Observations include three camera views and a 14-dimensional proprioceptive state, with a 14-dimensional action space. Actions and proprioceptive states are normalized using dataset quantiles. The policy predicts and executes 32-step action chunks.

In Stage 2, $\lambda_{\mathrm{WAM}}$ warms up to 1.0 over 1,000 steps. We keep $\lambda_{\mathrm{align}} = 1.0$ for the first 5k steps, decay it to 0.1 using a cosine schedule over the next 15k steps, and hold it constant thereafter. We evaluate all 50 tasks using 100 episodes per task under each of the clean and randomized settings. Success rates are averaged over tasks, and the overall score is the mean across both settings. Table 6 summarizes the training configuration.

Table 6: **Training details of WAVE-0 on RoboTwin 2.0.**

| Configuration | Stage 1: World Model | Stage 2: Action Policy |
|---|---|---|
| Number of GPUs | 8 | 8 |
| Precision | bf16 | bf16 |
| Optimizer | AdamW | AdamW |
| Global batch size | 512 | 512 |
| Peak learning rate | $1 \times 10^{-4}$ | $1 \times 10^{-4}$ |
| Final learning rate | $2.5 \times 10^{-6}$ | $1 \times 10^{-6}$ |
| Weight decay | $1 \times 10^{-2}$ | $1 \times 10^{-2}$ |
| Warmup steps | 1,000 | 1,000 |
| Gradient clipping | 1.0 | 1.0 |
| Training steps | 50,000 | 100,000 |
| Action chunk size $H$ | 32 | 32 |
| World model horizon $h$ | 31 | 31 |
| Latent action queries | – | 32 |
| World model | Trainable | Frozen |

### A.2 Experimental Details on Real-World Tasks

**Real-robot experiments.** We evaluate $\pi_{0.5}$, WAVE-0, LaWAM, and ImageWAM on a dual-arm Piper robot across three real-world manipulation tasks: Book Organization, USB Storage, and Keyboard Pushback. Book Organization requires placing a book into the designated book organizer, USB Storage requires transferring a USB drive from the tabletop into a designated storage container, and Keyboard Pushback requires pushing the keyboard back into the drawer. Figure 5 shows representative rollouts of WAVE-0. The complete real-robot dataset contains 600 demonstrations, with 200 demonstrations per task, comprising 310,099 frames recorded at a nominal frequency of 15 Hz. Each observation includes RGB images from one fixed head camera and two wrist cameras, together with a 14-dimensional proprioceptive state. Actions are represented as 14-dimensional absolute joint commands. Each task-specific policy is trained on its corresponding 200 demonstrations. Training is performed on eight NVIDIA A800 GPUs with 80 GB of memory per GPU, using distributed data parallelism, BF16 mixed precision, and the AdamW optimizer. During closed-loop evaluation, $\pi_{0.5}$ and WAVE-0 predict 32-step action chunks, whereas LaWAM and ImageWAM output 18-step and 16-step action chunks, respectively. $\pi_{0.5}$ and WAVE-0 execute 10 actions per policy inference, while LaWAM and ImageWAM execute 8.

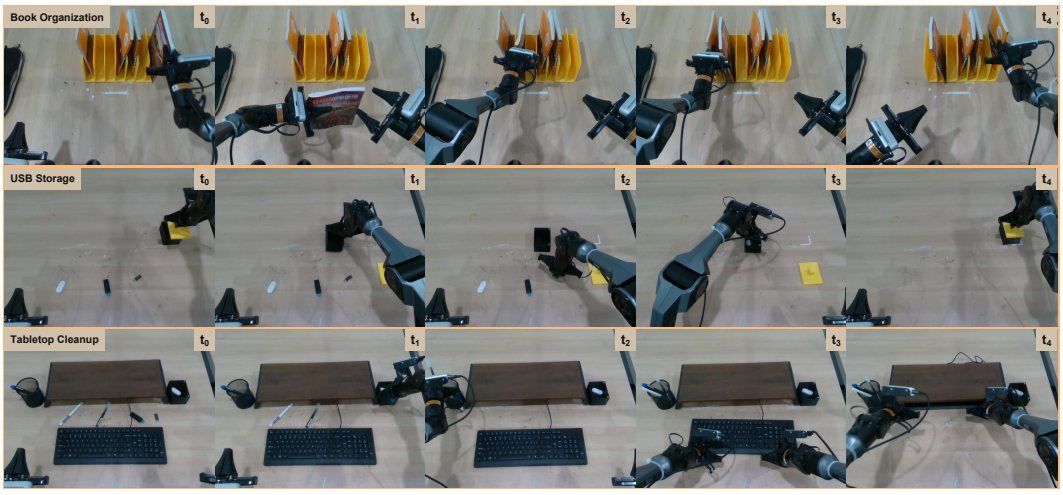



Figure 5: **Representative rollouts of WAVE-0 on three real-world manipulation tasks.**

$\pi_{0.5}$**.** For each real-robot task, we independently fine-tune $\pi_{0.5}$ on the corresponding 200 demonstrations for 30k training steps, using a global batch size of 256. We maintain an exponential moving

average of the policy parameters with a decay rate of 0.99 and use the resulting EMA checkpoints for real-robot evaluation. The policy predicts 32-step action chunks and executes 10 actions before replanning. The optimization configuration is summarized in Table 7.

Table 7: **Training details of $\pi_{0.5}$ in the real-world experiments.**

| **Configuration** | $\pi_{0.5}$ |
|---|---|
| Number of GPUs | 8 |
| Precision | bf16 |
| Optimizer | AdamW |
| Global batch size | 256 |
| Peak learning rate | $2.5 \times 10^{-5}$ |
| Final learning rate | $2.5 \times 10^{-6}$ |
| Weight decay | $1 \times 10^{-10}$ |
| Warmup steps | 1,500 |
| Gradient clipping | 1.0 |
| Training steps | 30,000 |
| Action chunk size $H$ | 32 |

**WAVE-0.** WAVE-0 is trained in two stages. In Stage 1, the latent world model is pretrained for 40k steps using all 600 real-robot demonstrations collected from the three manipulation tasks. In Stage 2, the pretrained world model is frozen, and a task-specific action policy is independently trained for 30k steps using the corresponding 200 demonstrations from each task. Both stages use linear learning-rate warmup followed by cosine decay. The world model predicts visual representations 32 steps into the future, while the action policy generates 32-step action chunks using 32 latent action queries. The detailed training configuration is summarized in Table 8.

Table 8: **Training details of WAVE-0 in the real-world experiments.**

| **Configuration** | **Stage 1: World Model** | **Stage 2: Action Policy** |
|---|---|---|
| Number of GPUs | 8 | 8 |
| Precision | bf16 | bf16 |
| Optimizer | AdamW | AdamW |
| Global batch size | 256 | 256 |
| Peak learning rate | $1 \times 10^{-4}$ | $5 \times 10^{-5}$ |
| Final learning rate | $2.5 \times 10^{-6}$ | $1 \times 10^{-6}$ |
| Weight decay | $1 \times 10^{-2}$ | $1 \times 10^{-2}$ |
| Warmup steps | 2,000 | 1,500 |
| Gradient clipping | 1.0 | 1.0 |
| Training steps | 40,000 | 30,000 |
| Action chunk size $H$ | 32 | 32 |
| World model horizon $h$ | 32 | 32 |
| Latent action queries | – | 32 |
| World model | Trainable | Frozen |

**LaWAM.** For each real-robot task, LaWAM is independently trained on the corresponding 200 demonstrations for 100k optimization steps. Training is performed on eight GPUs with a per-GPU batch size of 16, resulting in a global batch size of 128. The learning rate is linearly warmed up for 500 steps and subsequently decayed using a cosine schedule. LaWAM uses three RGB camera views as input and does not condition its flow-matching action head on the robot state. Its internal flow horizon is 50, from which the first 18 valid actions are returned during inference. The main training parameters are summarized in Table 9.

**ImageWAM.** For each real-robot task, ImageWAM is independently trained on the corresponding 200 demonstrations for 60k optimization steps. Training is performed on eight GPUs with a per-GPU batch size of 16, resulting in a global batch size of 128. We train the action prediction module using BF16 mixed precision and the AdamW optimizer. The learning rate is linearly warmed up for

3,000 steps, corresponding to 5% of the total training budget, and then decayed using a cosine schedule. ImageWAM jointly learns visual dynamics and action prediction from the three camera views and produces a 16-step absolute joint-action chunk during inference. The main training parameters are summarized in Table 10.

Table 9: **Training details of LaWAM in the real-world experiments.**

| Configuration | LaWAM |
|---|---|
| Number of GPUs | 8 |
| Precision | bf16 |
| Optimizer | AdamW |
| Per-GPU batch size | 16 |
| Global batch size | 128 |
| Peak learning rate | $1 \times 10^{-4}$ |
| Final learning rate | $5 \times 10^{-7}$ |
| Weight decay | $1 \times 10^{-8}$ |
| Warmup steps | 500 |
| Gradient clipping | 1.0 |
| Training steps | 100,000 |
| Internal flow horizon | 50 |
| Output action horizon $H$ | 18 |

Table 10: **Training details of ImageWAM in the real-world experiments.**

| Configuration | ImageWAM |
|---|---|
| Number of GPUs | 8 |
| Precision | bf16 |
| Optimizer | AdamW |
| Per-GPU batch size | 16 |
| Global batch size | 128 |
| Peak learning rate | $1 \times 10^{-4}$ |
| Final learning rate | $1 \times 10^{-6}$ |
| Weight decay | $1 \times 10^{-2}$ |
| Warmup steps | 3,000 |
| Gradient clipping | 1.0 |
| Training steps | 60,000 |
| Action chunk size $H$ | 16 |

**Inference efficiency.** Our lightweight latent world model adds only modest inference overhead to $\pi_{0.5}$. As shown in Table 11, WAVE-0 increases the mean server-side latency from 287.1 ms to 305.9 ms, an additional 18.8 ms, while enabling explicit future conditioning during action generation. Compared with ImageWAM, an efficient visual generation-based WAM requiring 582.6 ms per request, WAVE-0 reduces latency by 47.5%.

LaWAM achieves a lower latency of 223.9 ms with a smaller model of 2.3B parameters, compared with 4.3B for WAVE-0. It also returns 18 actions per inference request, whereas WAVE-0 predicts 32. These differences in model size and inference configuration should be considered when interpreting the latency comparison.

Table 11: **Mean server-side inference latency in the real-world experiments.**

| Model | Checkpoint | Mean server latency |
|---|---|---|
| $\pi_{0.5}$ | 30k | 287.1 ms |
| LaWAM | 100k | **223.9 ms** |
| ImageWAM | 60k | 582.6 ms |
| WAVE-0 | 30k | 305.9 ms |

### A.3 LIBERO-PRO DETAILED RESULTS

Table 12: **Results on LIBERO-Pro across different perturbation categories.**

| Method | Avg. | Spatial | | | | | Object | | | | | Goal | | | | | Long | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Obj. | Pos. | Sem. | Task | Env. | Obj. | Pos. | Sem. | Task | Env. | Obj. | Pos. | Sem. | Task | Env. | Obj. | Pos. | Sem. | Task | Env. |
| OpenVLA | 52.0 | 97.0 | 0.0 | 97.0 | 0.0 | **89.0** | **98.0** | 0.0 | 98.0 | 0.0 | 0.0 | 96.0 | 0.0 | 98.0 | 0.0 | **98.0** | 81.0 | 0.0 | 96.0 | 0.0 | **85.0** |
| $\pi_0$ | 44.0 | 95.0 | 0.0 | 97.0 | 0.0 | 60.0 | 94.0 | 0.0 | 90.0 | 0.0 | 29.0 | 94.0 | 0.0 | 93.0 | 0.0 | 39.0 | 79.0 | 0.0 | 82.0 | 0.0 | 27.0 |
| $\pi_{0.5}$ | 53.0 | 97.0 | 20.0 | 97.0 | 0.01 | 46.0 | **98.0** | 0.2 | 96.0 | 0.0 | 0.7 | **97.0** | **38.0** | 97.0 | 0.0 | 46.0 | **92.0** | 0.1 | 93.0 | 0.0 | 46.0 |
| Fast-WAM | 40.2 | 87.0 | 0.0 | 58.0 | 32.2 | 36.2 | 93.2 | 0.8 | 98.6 | 9.2 | 35.4 | 56.6 | 0.2 | 92.2 | 8.6 | 7.6 | 52.4 | 0.0 | 79.4 | 9.6 | 45.8 |
| ImageWAM | 48.7 | 95.6 | 18.0 | 95.2 | 49.6 | 56.6 | 91.0 | 0.0 | 99.6 | 6.6 | 40.6 | 68.2 | 3.4 | 98.2 | 10.0 | 31.4 | 53.6 | 0.0 | 96.2 | 10.0 | 50.2 |
| LaWAM | 52.7 | 97.8 | 28.8 | 93.2 | **52.6** | 43.4 | 91.8 | 10.6 | 99.6 | 10.0 | 61.0 | 68.4 | 9.0 | **98.4** | **28.4** | 25.6 | 62.8 | 1.2 | 96.6 | 23.0 | 51.2 |
| VLA-JEPA | 48.3 | 96.6 | 13.4 | 93.0 | 51.4 | 22.6 | 90.2 | 0.00 | **99.8** | 10.0 | 49.8 | 77.4 | 2.20 | 98.2 | 12.0 | 35.4 | 53.2 | 0.2 | 91.0 | 9.2 | 59.6 |
| JEPA-WAM | 41.9 | 85.4 | 0.2 | 69.0 | 51.6 | 35.0 | 92.8 | 0.0 | 99.0 | 10.0 | 14.0 | 57.8 | 0.0 | 72.6 | 11.2 | 46.6 | 36.8 | 0.0 | 85.4 | 9.4 | 61.6 |
| **WAVE-0 (Ours)** | **61.4** | **98.0** | **54.6** | **97.4** | 48.0 | 64.2 | 83.0 | **54.2** | **99.8** | **13.2** | **70.4** | 85.6 | 23.4 | 97.8 | 21.4 | 47.0 | 70.2 | **11.4** | **97.2** | **27.6** | 63.4 |

### A.4 ROBOTWIN DETAILED RESULTS

Table 13: **Per-task success rates (%) on RoboTwin 2.0.**

| Task | WAVE-0 (Ours) | | $\pi_{0.5}$ | | Motus | | Fast-WAM | | LingBot-VA | | LaWAM | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Clean | Rand. | Clean | Rand. | Clean | Rand. | Clean | Rand. | Clean | Rand. | Clean | Rand. |
| Adjust Bottle | **100** | **100** | **100** | 99 | 89 | 93 | **100** | **100** | 90 | 94 | **100** | **100** |
| Beat Block Hammer | **100** | **99** | 92 | 94 | 95 | 88 | 99 | 97 | 96 | 98 | 90 | 93 |
| Blocks Ranking RGB | **100** | 99 | 92 | 90 | 99 | 97 | **100** | **100** | 99 | 98 | 97 | **100** |
| Blocks Ranking Size | 90 | 87 | 78 | 74 | 75 | 63 | **94** | **98** | **94** | 96 | 93 | 89 |
| Click Alarmclock | **100** | 99 | **100** | 99 | **100** | **100** | **100** | **100** | 99 | **100** | **100** | **100** |
| Click Bell | **100** | **100** | **100** | 98 | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** |
| Dump Bin Bigbin | 95 | **98** | 86 | 91 | 95 | 91 | **97** | 96 | 89 | 96 | **97** | 95 |
| Grab Roller | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** |
| Handover Block | 97 | **91** | 59 | 45 | 86 | 73 | 95 | 81 | **99** | 78 | 96 | 87 |
| Handover Mic | 97 | 99 | 90 | 88 | 78 | 63 | **99** | **100** | 94 | 96 | 93 | 98 |
| Hanging Mug | 47 | 49 | 40 | 49 | 38 | 38 | **58** | **62** | 40 | 28 | 51 | 43 |
| Lift Pot | **100** | **100** | 93 | 88 | 96 | 99 | **100** | **100** | **100** | 99 | **100** | 99 |
| Move Can Pot | 91 | 94 | 72 | 72 | 34 | 74 | 90 | 88 | 94 | **97** | **98** | 93 |
| Move Pillbottle Pad | 98 | 97 | 74 | 80 | 93 | 96 | **100** | **99** | 99 | **99** | 97 | 90 |
| Move Playingcard Away | **100** | 99 | 97 | 93 | **100** | 96 | **100** | **100** | **100** | 99 | **100** | **100** |
| Move Stapler Pad | 55 | 52 | 42 | 27 | 83 | 85 | 77 | 64 | 91 | 79 | **94** | **87** |
| Open Laptop | 97 | 99 | 99 | 98 | 95 | 91 | 98 | **100** | 92 | 94 | **100** | **100** |
| Open Microwave | 58 | 74 | 28 | 14 | **95** | **91** | 62 | 45 | 82 | 86 | 41 | 43 |
| Pick Diverse Bottles | 78 | 81 | 82 | 78 | 90 | **91** | 80 | 85 | 89 | 82 | **91** | 88 |
| Pick Dual Bottles | 94 | 96 | 97 | 94 | 96 | 90 | **100** | 96 | **100** | **99** | **100** | 95 |
| Place A2B Left | 91 | **94** | 93 | 90 | 88 | 79 | 95 | 93 | 97 | 93 | **98** | 91 |
| Place A2B Right | **99** | 95 | 97 | 92 | 91 | 87 | 93 | **99** | 97 | 95 | 89 | 94 |
| Place Bread Basket | 87 | 91 | 76 | 80 | 91 | 94 | 91 | 93 | **97** | **95** | 92 | 85 |
| Place Bread Skillet | 90 | 85 | 80 | 73 | 86 | 83 | 90 | **93** | **95** | 90 | 90 | 83 |
| Place Burger Fries | **98** | 94 | 90 | 90 | **98** | 98 | 96 | **99** | 97 | 95 | 93 | 96 |
| Place Can Basket | 84 | 76 | 56 | 58 | 81 | 76 | 71 | 69 | 81 | **84** | **92** | 65 |
| Place Cans Plasticbox | 97 | **99** | 79 | 76 | 98 | 94 | 99 | 96 | **100** | **99** | **100** | 95 |
| Place Container Plate | 97 | 99 | 93 | 96 | 98 | 99 | 96 | **100** | 99 | 97 | **100** | **100** |
| Place Dual Shoes | 94 | **97** | 87 | 80 | 93 | 87 | 94 | 88 | 94 | 89 | **98** | 94 |
| Place Empty Cup | **100** | **100** | 98 | 96 | 99 | 98 | **100** | **100** | **100** | **100** | 99 | **100** |
| Place Fan | 91 | 92 | 69 | 59 | 91 | 87 | 96 | **96** | **99** | 93 | 92 | 93 |
| Place Mouse Pad | 90 | 85 | 71 | 73 | 66 | 68 | 83 | 89 | **93** | **96** | 91 | 84 |
| Place Object Basket | 91 | 89 | 77 | 70 | 81 | 87 | 89 | 88 | 91 | 88 | **92** | **90** |
| Place Object Scale | 95 | 95 | 81 | 78 | 88 | 85 | 90 | **97** | **96** | 95 | 95 | 88 |
| Place Object Stand | 93 | 95 | 81 | 83 | 98 | **97** | 90 | 94 | **99** | 96 | 92 | 93 |
| Place Phone Stand | **97** | 97 | 92 | 90 | 87 | 86 | **97** | **99** | **97** | 97 | 93 | 94 |
| Place Shoe | 99 | **100** | 97 | 98 | 99 | 97 | 96 | 99 | 98 | 98 | **100** | **100** |
| Press Stapler | 93 | 96 | **98** | **99** | 93 | 98 | 90 | 97 | 85 | 82 | **98** | 97 |
| Put Bottles Dustbin | 90 | 85 | 72 | 61 | 81 | 79 | **95** | 90 | 87 | 91 | 94 | **92** |
| Put Object Cabinet | 86 | 83 | 77 | 74 | 88 | 71 | **94** | **89** | 85 | 87 | 90 | 82 |
| Rotate QRcode | 93 | **95** | 80 | 85 | 89 | 73 | 93 | 89 | **96** | 91 | 94 | 89 |
| Scan Object | 95 | **95** | 91 | 88 | 67 | 66 | 89 | 92 | **96** | 91 | **96** | 90 |
| Shake Bottle | **100** | 99 | 98 | **100** | **100** | 97 | **100** | **100** | **100** | 97 | **100** | **100** |
| Shake Bottle Horizontally | **100** | **100** | 98 | **100** | **100** | 98 | **100** | **100** | **100** | 99 | **100** | **100** |
| Stack Blocks Three | 97 | 94 | 77 | 68 | 91 | 95 | 95 | 97 | **99** | **98** | 90 | 75 |
| Stack Blocks Two | **100** | **100** | 93 | 94 | **100** | 98 | **100** | **100** | **100** | 98 | **100** | 97 |
| Stack Bowls Three | 82 | 81 | 48 | 49 | 79 | **87** | 80 | 81 | 86 | 83 | **90** | 80 |
| Stack Bowls Two | 93 | 94 | 84 | 81 | 98 | 98 | 92 | 98 | 94 | 98 | **100** | **99** |
| Stamp Seal | 84 | 79 | 35 | 56 | 93 | 92 | 90 | 94 | **96** | **97** | 89 | 88 |
| Turn Switch | 70 | 77 | 67 | 74 | **84** | **78** | 61 | 59 | 44 | 45 | 47 | 56 |
| **Average** | 91.46 | 91.48 | 81.12 | 79.64 | 88.66 | 87.02 | 91.88 | **91.78** | **92.90** | 91.50 | 92.64 | 89.80 |