HARMOWAM: HARMONIZING GENERALIZABLE AND PRECISE MANIPULATION VIA ADAPTIVE WORLD ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs) have emerged as a promising paradigm for robot control by modeling physical dynamics. Current WAMs generally follow two paradigms: the "Imagine-then-Execute" approach, which uses video prediction to infer actions via inverse dynamics, and the "Joint Modeling" approach, which jointly models actions and video representations. Based on systematic experiments, we observe a fundamental trade-off between these paradigms: the former explicitly leverages world models for generalizable transit but lacks interaction precision, whereas the latter enables fine-grained, temporally coherent action generation but is constrained by the exploration space of the training distribution. Motivated by these findings, we propose **HarmoWAM**, an end-to-end WAM that fully leverages a world model to unify predictive and reactive control, enabling both generalizable transit and precise manipulation. Specifically, the world model provides spatio-temporal physical priors that condition two complementary action experts: a *predictive expert* that leverages latent dynamics for iterative action generation, and a *reactive expert* that directly infers actions from predicted visual evolution. To enable adaptive coordination, a Process-Adaptive Gating Mechanism is proposed to automatically determine the timing and location of switching between them. This allows the world model to drive the reactive expert to expand the exploration space and the predictive expert to perform precise interactions across different task stages. For evaluation, we construct three training-unseen test environments across six real-world robotic tasks, covering variations in background, position, and object semantics. Notably, HarmoWAM achieves strong zero-shot generalization across these scenarios, significantly outperforming prior state-of-the-art VLA models and WAMs by margins of 33% and 29%, respectively.

1 INTRODUCTION

Robust robotic manipulation requires policies that not only understand task instructions but also capture the underlying physical dynamics of the environment. **World Action Models (WAMs)** have recently emerged as a promising paradigm toward this goal, leveraging future visual prediction to provide spatio-temporal physical priors for action generation (Kim et al., 2026; Jang et al., 2025; Yuan et al., 2026). Existing WAMs can be broadly categorized into two paradigms (Yuan et al., 2026): (1) **Imagine-then-Execute**, which first generates imagined future trajectories and then infers actions via inverse dynamics (Feng et al., 2025; Jang et al., 2025; Chi et al., 2025); and (2) **Joint Modeling**, which directly models the joint distribution of actions and video representations, leveraging latent features from world models to guide action prediction (Kim et al., 2026; Bi et al., 2025; Zhu et al., 2025). However, we observe that these two paradigms exhibit markedly different behaviors across real-world tasks.

Driven by the inherent differences between these paradigms, we conduct a systematic empirical study on two representative real-world manipulation tasks, including dual-arm collaboration and precision stacking. For each task, we decompose the process into a transit phase between objects and an interaction phase, and evaluate performance under both In-Domain (ID) and three Out-of-Domain (OOD) settings, covering variations in background, object position, and object instances. We benchmark two representative approaches: Imagine-then-Execute (Wan2.2-TI2V-5B + AnyPos,

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.

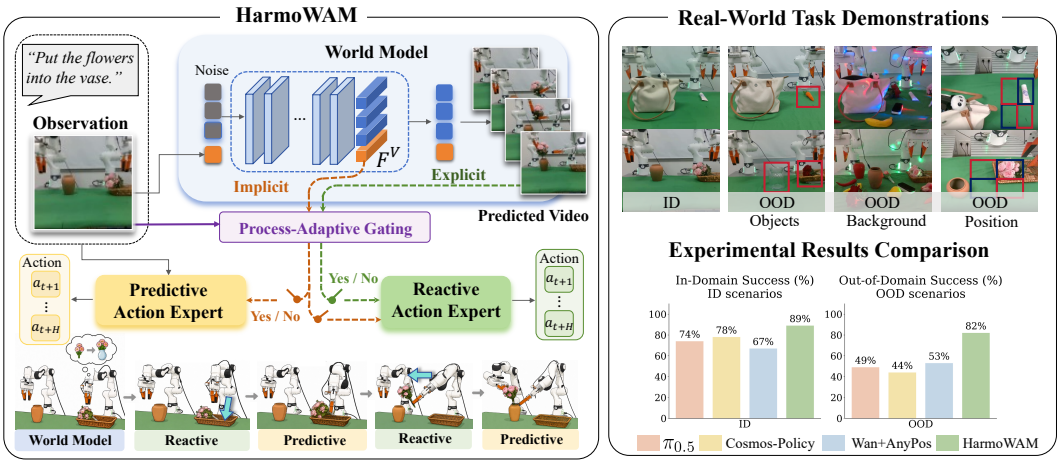


Figure 1: **Overview.** We propose HarmoWAM, an end-to-end WAM that jointly achieves generalizable transit and precise manipulation through a world model that provides physical dynamics priors and adaptively coordinates a predictive action expert and a reactive action expert. HarmoWAM achieves SOTA performance in ID settings and exhibits a substantial advantage in OOD scenarios.

following Chi et al. (2025); Tan et al. (2025)) and Joint Modeling (Wan2.2-TI2V-5B + Action DiT, following Jia et al. (2026); Hu et al. (2025)). Further details are provided in Section 3.2. We observe a fundamental trade-off between these paradigms: (1) **Imagine-then-Execute** leverages the generalization capability of world models, achieving near-perfect success in object approach even under OOD conditions, but lacks the precision required for fine-grained manipulation (with an average interaction success rate of only 60% across ID and OOD settings); (2) **Joint Modeling** achieves stable control precision above 80% when initialized near target objects, but fails to reliably approach them in OOD settings (with success rates below 60%), highlighting an exploration space constrained by the supervised fine-tuning (SFT) data distribution.

Motivated by these findings, we introduce **HarmoWAM**, an end-to-end WAM that unifies generalizable transit and precise manipulation (as illustrated in Figure 1). In this design, a world model generates future visual observations and latent representations as spatio-temporal conditions to adaptively coordinate predictive and reactive control across two complementary action experts. Both action experts are built upon Transformer-based backbones. A predictive action expert leverages latent video features to guide fine-grained, temporally coherent action generation. Meanwhile, a reactive expert directly infers actions from predicted future observations and corresponding latent features. To enable adaptive coordination, we propose a Process-Adaptive Gating Mechanism that dynamically determines when and where to switch between two action experts. Specifically, following a keyframe extraction strategy (Shridhar et al., 2023), we automatically construct a process-aware dataset to train a gating network that predicts the current task stage (i.e., transit or interaction) from visual observations. This enables HarmoWAM to leverage world model guidance for generalizable exploration through the reactive expert and precise interaction through the predictive expert, forming a unified closed-loop control framework.

To comprehensively evaluate HarmoWAM's efficacy, we conduct experiments across six real-world tasks (four single-arm, two dual-arm) under ID and three distinct OOD scenarios. As shown in Figure 1, HarmoWAM establishes a new state-of-the-art (SOTA) by outperforming the leading VLA and WAMs methods by 15% and 11% in ID settings, respectively. This performance gap further widens in OOD scenarios, where HarmoWAM demonstrates strong zero-shot generalization, outperforming state-of-the-art VLA and WAM methods by 33% and 29% on average across all variation types. HarmoWAM runs at approximately 4 Hz model inference frequency, with an action-output throughput of 48 predicted actions/s using 12-step action chunks. Our real-world evaluations further demonstrate that HarmoWAM maintains generalizable transit alongside high-precision manipulation across long-horizon tasks. Our main contributions are as follows:

1) We propose HarmoWAM, a novel end-to-end WAM that leverages spatio-temporal conditions provided by a world model to unify predictive action generation and reactive execution, effectively reconciling generalization and precision in robotic manipulation.

2) We introduce a Process-Adaptive Gating Mechanism that enables stage-aware coordination in HarmoWAM, dynamically routing control under world model guidance, the reactive expert expands the exploration space, while the predictive expert ensures precise manipulation.
3) We conduct extensive real-world evaluations across diverse manipulation tasks and OOD scenarios, showing that HarmoWAM not only achieves strong zero-shot generalization but also maintains robust performance in long-horizon settings.

2 Related Work

Vision-language-action (VLA) models (Brohan et al., 2022; Kim et al., 2024; Chen et al., 2024; Liu et al., 2026; Li et al., 2024a; Lin et al., 2025; Pertsch et al., 2025) adapt pretrained vision-language models (VLMs) (Karamcheti et al., 2024; Alayrac et al., 2022) to embodied control. Early efforts focused on scaling robot demonstrations and transferring pretrained VLM knowledge to action learning (Zitkovich et al., 2023; Team et al., 2024; Li et al., 2024b; Gu et al., 2025). More recent work strengthens action generation by replacing discrete or deterministic heads with continuous generative policies for precise control (Wen et al., 2024; 2025; Liu et al., 2024; 2025; Kim et al., 2025; Chen et al., 2025a). In particular, flow-based formulations reduce sampling overhead while maintaining action fidelity (Black et al., 2024; Intelligence et al., 2025; Liu et al., 2026; Intelligence et al., 2026). Despite these advances, VLA models remain constrained by their training distribution, with broader generalization often requiring costly additional data.

World Action Models (WAMs) explore video generation and predictive visual modeling for robot control (Zhou et al., 2025; Bharadhwaj et al., 2024; Ye et al., 2026a; Du et al., 2023; Wu et al., 2023; Zhou et al., 2024; Won et al., 2025; Cheang et al., 2024; Cen et al., 2025a;b; Zheng et al., 2025; Ye et al., 2026b) through two broad paradigms. *Imagine-then-Execute* methods predict future observations and infer actions from imagined frames (Feng et al., 2025). Examples include DreamGen (Jang et al., 2025) and WoW (Chi et al., 2025), which use video prediction to capture environment dynamics for downstream control. *Joint Modeling* couples video modeling with action learning (Kim et al., 2026; Pai et al., 2025; Li et al., 2026), incorporating predictive visual information as latent supervision or action conditioning. VPP (Hu et al., 2025), Video2Act (Jia et al., 2026), Genie Envisioner (Liao et al., 2025), and Video Policy (Liang et al., 2025) condition action generation on video-model representations. UVA (Li et al., 2025), UWM (Zhu et al., 2025), Cosmos-Policy (Kim et al., 2026), and Motus (Bi et al., 2025) jointly model video and actions within unified generative frameworks, while Fast-WAM (Yuan et al., 2026) retains benefits from video modeling without explicit future generation at inference. In this paper, we propose HarmoWAM, an end-to-end WAM that fully exploits both explicit and implicit spatio-temporal priors from a world model to unify generalizable transit and precise manipulation.

3 Methods

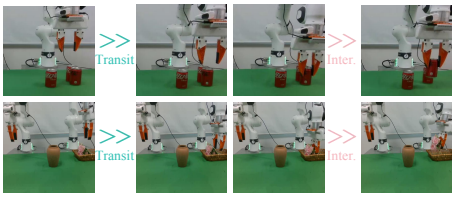
3.1 Preliminaries

Action Prediction Problem Formulation. We formulate robotic manipulation as a probabilistic sequential decision-making problem (Black et al., 2024). At each timestep t, conditioned on a natural language instruction $\mathbf{l}_t$ and a visual observation $\mathbf{I}_t \in \mathbb{R}^{H\times W\times 3}$, the policy π_θ predicts a future action sequence of length H, defined as $\mathbf{a}_{t+1:t+H} \sim \pi_\theta(\cdot \mid \mathbf{I}_t, \mathbf{l}_t)$. For single-arm manipulation, the 7-DoF action $\mathbf{a} \in \mathbb{R}^7$ comprises relative position (x, y, z), Euler rotation $(\mathrm{roll}, \mathrm{pitch}, \mathrm{yaw})$, and binary gripper state $g \in \{0, 1\}$. For dual-arm settings, concatenating both arms' control vectors yields a 14-DoF action space. Appendix D.1 provides preliminaries on Inverse Dynamics Model (IDM)-based action inference.

3.2 Motivation

While existing WAMs, whether based on the Imagine-then-Execute or Joint Modeling paradigm, have demonstrated promising capabilities, our real-world experiments show that their representative implementations exhibit distinct limitations across task stages. Therefore, we conduct systematic real-world evaluations on representative baselines to characterize their manipulation boundaries and identify their phase-dependent strengths and limitations.

Table 1: **Motivation.** Comparison of *Imagine-then-Execute* and *Joint Modeling* under ID and OOD settings. Interaction performance (*) uses near-target initialization for Joint Modeling to isolate manipulation precision from transit failures. Imagine-then-Execute requires no reinitialization, as it autonomously reaches the target vicinity in all evaluated settings.



Method	Domain	Gen Scenarios	Stack Coke Cans		Put Flowers in Vase	
			Transit	Interaction	Transit	Interaction
Imagine-then-Execute	In	—	10/10	7/10	10/10	8/10
	OOD	Background	10/10	6/10	10/10	6/10
		Position	10/10	5/10	10/10	2/10
		Objects	10/10	7/10	10/10	7/10
Joint Modeling	In	—	9/10	9/10*	9/10	10/10*
	OOD	Background	5/10	8/10*	5/10	9/10*
		Position	3/10	10/10*	0/10	10/10*
		Objects	0/10	10/10*	6/10	10/10*

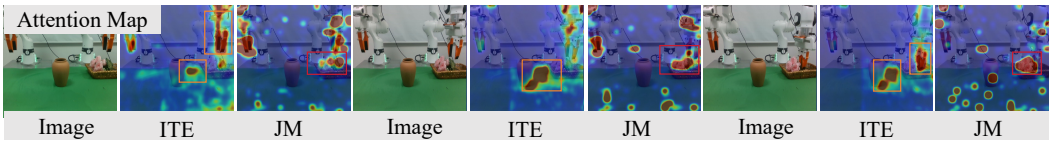


Figure 2: **Attention visualizations.** The figure presents attention maps from the final Transformer layers of the Imagine-then-Execute (ITE) and Joint Modeling (JM) baselines on *Put Flowers in Vase*.

Experimental Setup. For the *Imagine-then-Execute* paradigm, following Chi et al. (2025); Mi et al. (2026), we instantiate the baseline using Wan2.2-TI2V-5B (Wan et al., 2025) for future-video generation and AnyPos (Tan et al., 2025) for inverse-dynamics-based action inference. For the *Joint Modeling* paradigm, following VPP (Hu et al., 2025), we integrate latent features from Wan2.2-TI2V-5B into an Action Diffusion Transformer (DiT) through cross-attention. For a fair comparison, both implementations share the same Wan2.2 checkpoint, training data, action space, and evaluation protocol, while preserving their paradigm-specific integration of world-model predictions into action generation. We conduct the motivation study on two representative real-world tasks: *Put Flowers in Vase* and *Stack Coke Cans*. We decompose each manipulation process into two phases: *Transit* denotes target-reaching or movement between objects, whereas *Interaction* denotes contact-sensitive manipulation. Both tasks are evaluated under in-domain (ID) settings and three out-of-domain (OOD) scenarios, covering variations in background, object position, and object semantics.

Findings and Analysis. In Table 1, the *Imagine-then-Execute* baseline achieves near-perfect success in the transit phase across both ID and OOD settings, demonstrating strong generalization in object-to-object motion. However, its average interaction success falls to 75% in ID and 55% in OOD settings, indicating limited fine-grained manipulation precision. In contrast, the *Joint Modeling* baseline achieves strong ID performance, averaging over 90% across the transit and interaction evaluations. However, its OOD transit success drops notably to 32%, revealing an exploration space constrained by the supervised fine-tuning (SFT) distribution. Notably, when initialized near the target object, it still achieves an average interaction success of 95% in OOD settings, suggesting that its limitation primarily lies in exploration rather than manipulation. To support our findings, we visualize the attention maps of the two baselines in Figure 2. Joint Modeling focuses more on manipulated objects, while Imagine-then-Execute attends to the gripper and task-relevant surroundings. These results reveal a consistent transit–interaction complementarity between the evaluated representative implementations: the *Imagine-then-Execute* excels at generalizable transit but lacks manipulation precision, whereas the *Joint Modeling* achieves high precision but fails to reliably explore and reach target objects in unseen environments. These findings motivate a unified WAM for both generalizable transit and precise manipulation. Further details are provided in Appendix D.2.

3.3 HarmoWAM Framework

Motivated by the above findings, we propose HarmoWAM, a novel WAM that jointly achieves generalizable transit and precise manipulation. Going beyond architectural refinements, our design represents a first step toward a unified paradigm that organically integrates these complementary capabilities, fully unlocking the knowledge encoded in world models.

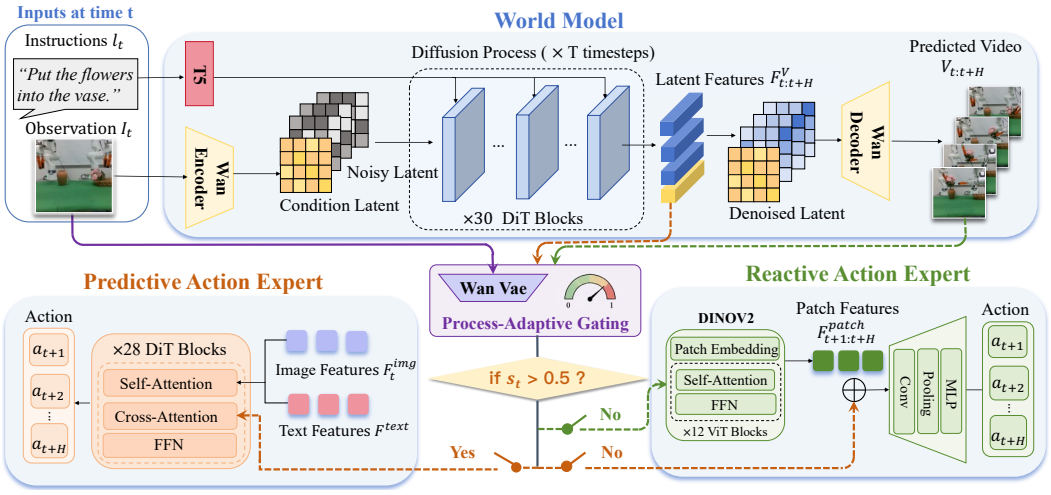


Figure 3: **Framework.** HarmoWAM adopts an adaptive framework that tightly integrates a generative world model with two complementary action experts. The world model provides both explicit future predictions and implicit latent representations. Conditioned on current latent features, the predictive expert generates structured actions for precise manipulation, while the reactive expert leverages future predicted frames and their latent features for generalizable transit. A Process-Adaptive Gating mechanism predicts the task stage from observations and dynamically routes control between the two action experts, resulting in a unified closed-loop control paradigm.

Overall Architecture. As shown in Figure 3, HarmoWAM adopts an adaptive architecture by leveraging a world model to seamlessly bridge spatio-temporal reasoning and motion generation with two complementary action experts. The world model is initialized from the official Wan2.2-TI2V-5B checkpoint (Wan et al., 2025) and predicts 13-frame videos in a rolling-window manner. The predicted videos and their spatio-temporal features provide visual dynamics priors for downstream action generation. To ground the world model's physical dynamics into actionable robot control, we design two complementary experts. Both action experts are built upon Transformer-based backbones. The predictive expert is instantiated as a 1B-parameter DiT, consisting of 28 Transformer blocks and incorporating a SigLIP (Zhai et al., 2023) image encoder and a text encoder (Raffel et al., 2020) to provide real-time, closed-loop visual observations and task instructions, enabling structured action generations. Meanwhile, the reactive expert is designed for real-time and interpretable action generation by directly leveraging the generalizable visual predictions from the world model. It integrates a DINOv2-base (Oquab et al., 2023) model with an Orientation Decoder ($\mathcal{D}_{ori}$), where the decoder is implemented using multi-scale convolutional layers (Tan et al., 2025) to map high-level visual features into the action space.

Coordination in HarmoWAM. We design a unified framework that tightly integrates a world model with complementary action experts to exploit its implicit latent representations and explicit video predictions. Given the current front-view observation and instruction, the world model uses 5 video-diffusion denoising steps to predict $\mathbf{V}_{t:t+H}$ and extracts latent representations $\mathcal{F}^{\mathbf{V}}_{t:t+H}$ from the final DiT block at the last denoising step as spatio-temporal conditions.

World Model to Predictive Expert. To fully exploit latent representations from the world model, which encode rich temporal dynamics critical for action generation, the predictive expert leverages the current-step latent feature $\mathcal{F}^{\mathbf{V}}_t \in \mathbb{R}^{B\times 80\times 3072}$ as an implicit condition. This feature is incorporated into the Action DiT through cross-attention layers, together with image features $\mathcal{F}^{img}_t$ and textual features $\mathcal{F}^{text}$. At each action diffusion denoising step k, the model predicts the added Gaussian noise $\epsilon_\theta = \mathcal{D}_{\theta_{\text{pred}}}(\mathbf{a}_{t+1:t+H}, \tau_k \mid \mathcal{F}^{img}_t, \mathcal{F}^{text}, \mathcal{F}^{\mathbf{V}}_t)$, where τ_k denotes the timestep embedding. This temporally coherent conditioning enables the expert to internalize physical dynamics during action prediction, leading to more structured and consistent decision-making.

World Model to Reactive Expert. Unlike prior IDM approaches that rely solely on observed video frames, which primarily provide low-level visual cues, we additionally incorporate the corresponding future latent dynamics. Concretely, for each future timestep $s \in \{t+1, \ldots, t+H\}$, the reactive expert takes both the predicted frame $\mathbf{V}_s$ and its latent representation $\mathcal{F}^{\mathbf{V}}_s \in \mathbb{R}^{B\times 80\times 3072}$ as inputs.

A DINOv2 encoder extracts patch-level geometric features $\mathcal{F}_s^{\text{patch}} \in \mathbb{R}^{B\times 1369\times 768}$ from $\mathbf{V}_s$. These features are concatenated with the latent features along the token dimension, after applying average pooling to reduce the channel dimension of $\mathcal{F}_s^{\mathbf{V}}$ from 3072 to 768, forming a fused representation: $\mathcal{F}_s^{\text{fuse}} = [\mathcal{F}_s^{\text{patch}}; \mathcal{F}_s^{\mathbf{V}}]$. The Orientation Decoder $\mathcal{D}_{\text{ori}}$ extracts a global representation to predict the action: $\hat{\mathbf{a}}_s = \mathcal{D}_{\text{ori}}(\mathcal{F}_s^{\text{fuse}})$. This design allows inverse dynamics to be performed not only based on low-level visual observations but also informed by high-level spatio-temporal knowledge.

3.4 Process-Adaptive Gating Mechanism

Motivated by the transit–interaction complementarity in Section 3.2, we propose a Process-Adaptive Gating Mechanism that uses real-time observations to dynamically switch between the two action experts for efficient coordination. Implemented as a lightweight MLP classifier, the gate uses features $\mathcal{F}_t^{\text{VAE}}$ extracted from the current observation $\mathbf{I}_t$ by the frozen Wan VAE encoder to predict a score $s_t \in [0, 1]$, indicating whether the state favors precise interaction or generalizable transit.

During supervised training, we follow a keyframe extraction pipeline (Shridhar et al., 2023) to automatically construct a process-aware dataset from robot proprioceptive signals. Interaction phases are identified by changes in gripper state or, for a few tasks without distinctive gripper-state transitions, by task-specific end-effector height thresholds, which indicate moments requiring precise manipulation. The ± 20 frames around each identified event are labeled as $y = 1$ for the predictive action expert. In contrast, frames without such interaction cues correspond to transit phases (e.g., approaching the target) and are labeled as $y = 0$ for the reactive action expert. These labels are used only for training and are not accessed during inference. More details are provided in Appendix D.3. The gating network is optimized using a Binary Cross-Entropy (BCE) loss function $\mathcal{L}_{gate}$ as follows:

$$\mathcal{L}_{gate} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(s_i) + (1 - y_i)\log(1 - s_i)\right], \tag{1}$$

where N is the batch size, $y_i \in \{0, 1\}$ is the label for frame i, and $s_i \in [0, 1]$ is the predicted probability of it being in an interaction phase. **During inference**, the gate selects the predictive expert when $s_t > 0.5$ and the reactive expert otherwise. This allows HarmoWAM to dynamically select the appropriate action expert, maintaining both generalization and precision throughout manipulation.

3.5 Downstream Training Recipe

To effectively align the learning of physical dynamics with downstream action generation, HarmoWAM is optimized through a two-stage training paradigm.

Stage 1: World Model Finetuning. We first adapt the official Wan2.2-TI2V-5B checkpoint to capture task-specific visual dynamics from real-world demonstrations. The model is fully finetuned with a conditional Flow Matching objective. Let $\mathbf{x}_1$ denote the clean video latent from a demonstration and $\mathbf{x}_0 \sim \mathcal{N}(0, I)$ denote a Gaussian noise latent. Given the condition $\mathbf{c}$, including the current observation and task instruction, we sample a flow interpolation variable $\xi \in [0, 1]$ and construct: $\mathbf{x}_\xi = (1 - \xi)\mathbf{x}_0 + \xi\mathbf{x}_1$. With target velocity $\mathbf{v}_\xi = \frac{d\mathbf{x}_\xi}{d\xi} = \mathbf{x}_1 - \mathbf{x}_0$, we minimize:

$$\mathcal{L}_{\text{stage1}} = \mathbb{E}_{\mathbf{x}_0,\mathbf{x}_1,\xi,\mathbf{c}}\left[w(\xi)\,\|f_\theta(\mathbf{x}_\xi, \xi, \mathbf{c}) - \mathbf{v}_\xi\|_2^2\right], \tag{2}$$

where $w(\xi)$ is the flow-step-dependent weighting function.

Stage 2: Action Experts Finetuning. We optimize the two experts and the gating network, conditioned on the explicit and implicit video features provided by the frozen world model. **The predictive expert** is trained using a standard diffusion denoising loss ($\mathcal{L}_{\text{pred}}$) to ensure accurate action generation, where ϵ denotes the Gaussian noise added to the action sequence. **The reactive expert** uses a Smooth L1 loss ($\mathcal{L}_{\text{react}}$) between predicted actions $\hat{\mathbf{a}}_t$ and demonstration actions $\mathbf{a}_t$.

$$\begin{aligned}\mathcal{L}_{\text{pred}} &= \mathbb{E}_{\mathbf{a}_{t+1:t+H},\epsilon\sim\mathcal{N}(0,1)}\left[\|\epsilon_\theta - \epsilon\|_2^2\right], \\ \mathcal{L}_{\text{react}} &= \mathbb{E}\left[d(\hat{\mathbf{a}}_{t+1:t+H}, \mathbf{a}_{t+1:t+H})\right].\end{aligned} \tag{3}$$

The overall objective is a weighted sum of the expert and gating losses:

$$\mathcal{L}_{\text{stage2}} = \mathcal{L}_{\text{pred}} + \lambda_{react}\mathcal{L}_{\text{react}} + \lambda_{\text{gate}}\mathcal{L}_{\text{gate}} \tag{4}$$

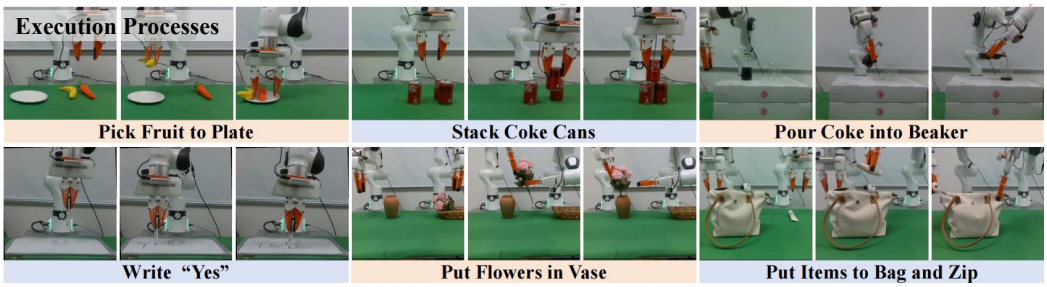


Figure 4: **Real-world execution.** Representative execution sequences across six real-world tasks.

Table 2: **In-domain quantitative results on real-world tasks.** Average cumulative stage success rates for four single-arm and two dual-arm tasks.

Method	Single-arm Tasks (Avg)				Dual-arm Tasks (Avg)		Avg
	Pick Fruit	Stack Cans	Pour Coke	Write "Yes"	Put Flowers	Put Items	
$\pi_{0.5}$	0.80	0.68	0.75	0.83	0.72	0.67	0.74
VPP	0.80	0.60	0.78	0.73	—	—	0.73
Wan+AnyPos	0.88	0.60	0.78	0.72	0.53	0.52	0.67
QwenVLA-OFT	0.78	0.30	0.73	0.72	—	—	0.63
Cosmos-Policy	0.93	0.65	0.80	0.83	0.75	0.72	0.78
Ours	**0.95**	**0.90**	**0.88**	**0.92**	**0.85**	**0.85**	**0.89**

where $\lambda_{react} = 0.1$ and $\lambda_{\text{gate}} = 0.05$ are used to balance the scales of different losses.

4 Experiments

4.1 Experiment Setup

Data Collection. We evaluate HarmoWAM on real-world manipulation tasks using Franka Research 3 robots, each equipped with three Intel RealSense cameras. We collect demonstrations for four single-arm tasks—*Pick Fruit to Plate*, *Stack Coke Cans*, *Pour Coke into Beaker*, and *Write "Yes"*, and two dual-arm collaborative tasks, *Put Flowers in Vase* and *Put Items to Bag and Zip*. Each task includes 100 demonstration trajectories collected via SpaceMouse teleoperation. Details are provided in Appendix A and B.

Training and Evaluation Details. We compare against five baselines spanning three categories: (1) SOTA VLA policies, including $\pi_{0.5}$ (Intelligence et al., 2025) and QwenVLA-OFT (Bai et al., 2025); (2) Imagine-then-Execute WAMs, instantiated using Wan2.2-TI2V-5B (Wan et al., 2025) for future video generation and AnyPos (Tan et al., 2025) for action inference, following Chi et al. (2025); and (3) Joint Modeling WAMs, including VPP (Hu et al., 2025) and Cosmos-Policy (Kim et al., 2026), with training details and additional descriptions provided in Appendix C.1. For complex dual-arm tasks, we compare only the strongest baseline in each category, as others perform markedly worse. Each method is evaluated over 20 independent episodes per task with randomized tabletop object positions. We report the average cumulative stage success rate, computed as the average over all critical sub-stages, with evaluation details and stage-wise results provided in Appendix F.1 and F.2.

4.2 In-Domain Results

As shown in Table 2, HarmoWAM achieves the best ID performance at 89%, outperforming $\pi_{0.5}$ (74%), Cosmos-Policy (78%), and Wan+AnyPos (67%). We additionally compute the standard deviation of our method ($\pm$ 3%) by repeating the evaluation three times under the same setting. HarmoWAM demonstrates strong advantages on tasks requiring high precision and long horizons. Specifically, on *Stack Coke Cans*, it achieves 90%, highlighting its ability to accurately align objects during stacking. HarmoWAM attains 85% on the long-horizon task *Put Items to Bag and Zip*, outperforming the next-best baseline by 13%, indicating improved robustness in both transit and interaction phases. It also achieves 85% on *Put Flowers in Vase*, demonstrating effective bimanual coordination during handover and insertion. These results suggest that our proposed framework effectively unlocks world model spatio-temporal knowledge, enabling coordinated control between

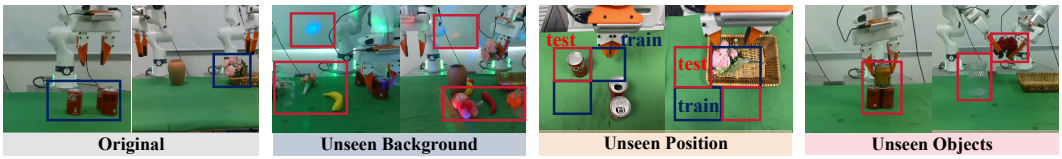


Figure 5: **Generalization experiments.** Red boxes highlight unseen objects, background variations, and manipulated object positions, while blue boxes indicate original training configurations.

Table 3: **Generalization performance.** Average cumulative stage success rates under three OOD settings. Global Avg. is the mean over the three OOD settings. The percentage drop (in parentheses) is relative to the ID success rate.

Method	Gen Scenario	Single-arm Tasks (Avg)				Dual-arm Tasks (Avg)		Avg	Global Avg
		Pick Fruit	Stack Cans	Pour Coke	Write "Yes"	Put Flowers	Put Items		
$\pi_{0.5}$	Background	0.65	0.40	0.63	0.67	0.62	0.60	0.60	0.49 (33.8% ↓)
	Position	0.20	0.08	0.43	0.55	0.37	0.26	0.32	
	Objects	0.55	0.43	0.53	0.63	0.60	0.49	0.54	
VPP	Background	0.38	0.25	0.68	0.40	—	—	0.43	0.41 (43.8% ↓)
	Position	0.28	0.10	0.20	0.33	—	—	0.23	
	Objects	0.68	0.40	0.68	0.50	—	—	0.57	
Wan+AnyPos	Background	0.58	0.43	0.65	0.65	0.47	0.38	0.53	0.53 (20.9% ↓)
	Position	0.83	0.40	0.33	0.50	0.45	0.45	0.49	
	Objects	0.63	0.65	0.68	0.68	0.40	0.43	0.58	
QwenVLA-OFT	Background	0.48	0.13	0.58	0.63	—	—	0.46	0.41 (34.9% ↓)
	Position	0.28	0.10	0.33	0.42	—	—	0.28	
	Objects	0.63	0.10	0.73	0.52	—	—	0.50	
Cosmos-Policy	Background	0.85	0.38	0.73	0.57	0.42	0.48	0.57	0.44 (43.6% ↓)
	Position	0.20	0.20	0.15	0.55	0.25	0.21	0.26	
	Objects	0.80	0.38	0.68	0.22	0.50	0.44	0.50	
Ours	Background	**0.90**	**0.78**	**0.83**	**0.82**	**0.73**	**0.78**	**0.81**	**0.82** (7.9% ↓)
	Position	**0.93**	**0.78**	**0.80**	**0.83**	**0.80**	**0.64**	**0.80**	
	Objects	**0.88**	**0.85**	**0.88**	**0.87**	**0.80**	**0.81**	**0.85**	

two experts for robust manipulation across diverse tasks. In Figure 4, we visualize real-world execution, with additional visualizations, failure cases, and videos provided in Appendix E.1, Appendix G, and the supplementary material.

4.3 Out-of-Domain Generalization

We evaluate zero-shot generalization under three OOD scenarios: unseen backgrounds, object positions, and manipulated objects. Representative examples are shown in Figure 5, while the complete examples and detailed OOD settings are provided in Appendix E.2 and C.3.

Unseen Background. As shown in Table 3, in cluttered OOD scenarios with unseen distractors and lighting changes, HarmoWAM maintains 81%, outperforming Wan+AnyPos (53%) and $\pi_{0.5}$ (60%). Compared methods often suffer from visual misgrounding, confusing distractors with targets during transit. In contrast, HarmoWAM leverages a world model to predict task-relevant visual evolution, providing generalizable semantic guidance for two action experts and enabling coordinated execution for reliable performance in visually complex OOD environments.

Unseen Position. We evaluate spatial generalization in regions beyond the training distribution, which constitute the most challenging OOD setting. In these regions, $\pi_{0.5}$ exhibits a significant drop to 32%, and Cosmos-Policy to 26%. The Imagine-then-Execute paradigm improves generalizable transit via video prediction, but its inverse-dynamics-based action inference still lacks precision. For example, even in the simple *Pick Fruit* task, the robot can reach the OOD region but fails to achieve accurate grasping. In contrast, HarmoWAM mitigates this trade-off through world-model-guided expert collaboration: the reactive expert generalizes beyond the SFT distribution during transit, while the predictive expert exploits temporally coherent dynamics for precise interaction.

Unseen Objects. We introduce unseen objects in each task to evaluate semantic generalization. All other methods exhibit significant performance degradation, whereas HarmoWAM achieves an 85% zero-shot success rate. This further demonstrates that our approach effectively leverages the semantic knowledge encoded in the world model, using both explicit and implicit conditioning to coordinate the two action experts for reliable object approach and interaction with unseen objects.

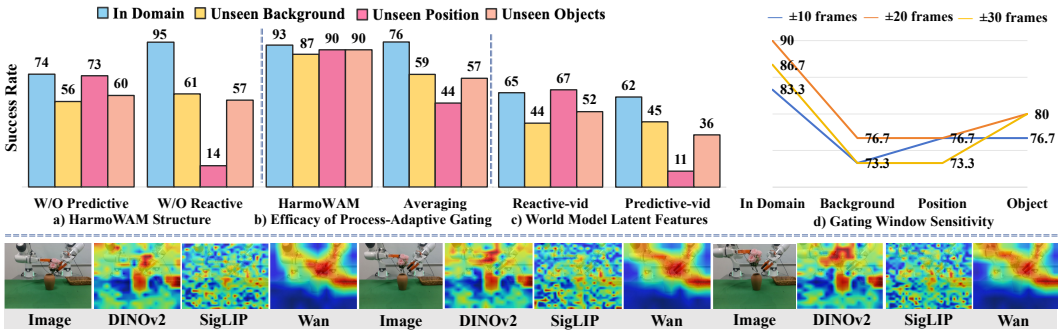


Figure 6: **Ablation study.** We investigate (a) HarmoWAM structure, (b) efficacy of Process-Adaptive Gating, and (c) impact of world model latent features on both action experts. The "-vid" suffix indicates that video latent features are excluded from the action expert's conditioning. Panel (d) evaluates gating window sensitivity on *Put Flowers in Vase*, with ± 20 frames as the default window. The lower part compares Grad-CAM activations of DINOv2 features, SigLIP features, and Wan world-model latent features across frames of *Put Flowers in Vase*.

4.4 Ablation Study

We conduct ablations on *Put Flowers in Vase* and *Pick Fruit to Plate* (Figure 6(a–c)), averaging performance across tasks. These tasks cover long-horizon execution, bimanual coordination, and precise manipulation. Each ablation variant is evaluated over 10 trials per task under each setting.

HarmoWAM Structure. Figure 6(a) validates HarmoWAM's architecture and both experts' necessity. Without the reactive expert, position-OOD success drops to 14%, indicating that the predictive expert alone struggles to translate world-model knowledge into actions that expand the exploration space. Without the predictive expert, success rates drop to 73% under position OOD and 60% under object-instance OOD. This suggests that reactive inference from predicted visual observations alone is insufficient for fine-grained actions under visual variations. Thus, world-model-guided expert coordination is essential for achieving better overall performance than either expert alone.

Efficacy of Process-Adaptive Gating. We compare Process-Adaptive Gating with Averaging, which averages both experts' outputs at each timestep. Averaging performs worse under OOD settings, with a 46-percentage-point drop relative to Process-Adaptive Gating in the position OOD scenario. These results demonstrate the efficacy of Process-Adaptive Gating in coordinating complementary experts for generalizable transit and precise interaction. On *Put Flowers in Vase*, changing the interaction-label window from ± 20 to ± 10 or ± 30 frames alters the average success rate by at most 3.4 percentage points (Figure 6(d)). Together with stage-recognition and closed-loop routing evaluations in Appendix D.3, these results further validate the gate's reliability and robustness.

World Model Latent Features. World-model video latent features are crucial to conditioning both action experts. Removing these features reduces the reactive expert's ID performance to 65% and OOD average to 54%, and the predictive expert's ID performance from 95% to 62%. The lower part of Figure 6 compares Grad-CAM activations of DINOv2, SigLIP, and Wan world-model latent features with respect to an action-prediction target. Compared with DINOv2 and SigLIP, Wan latent features exhibit more concentrated and temporally coherent responses around the moving grippers and flower, suggesting their relevance to action prediction and providing qualitative support for the ablation results. See Appendix F.3 for world-model denoising-step ablations.

5 Conclusion

In this work, we identify a fundamental trade-off between generalizable transit and precise manipulation in existing World Action Models (WAMs). We propose HarmoWAM, an end-to-end WAM that unifies predictive and reactive control under a shared world model. By leveraging spatio-temporal priors and coordinating complementary experts via Process-Adaptive Gating, HarmoWAM enables stage-aware control for both broad exploration and fine-grained interaction. Extensive real-world experiments across diverse tasks and unseen environments show that HarmoWAM consistently outperforms prior approaches with strong zero-shot generalization and high manipulation accuracy.

AI USE STATEMENT

We used generative AI tools to assist with language polishing and literature searches. The authors take responsibility for the research methodology, code, experimental results, and analyses.

REPRODUCIBILITY STATEMENT

To support reproducibility, we describe the model architecture and training objectives in Section 3 and the evaluation protocol in Section 4.1. Appendices A and B document the robot platforms, demonstration collection, and task definitions. Appendices C and D provide additional training, inference, and method details, including the gating-label construction procedure. Metric definitions, stage-wise results, and additional ablations are provided in Appendix F.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 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. *arXiv preprint arXiv:2512.13030*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

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.

Jun Cen, Siteng Huang, Yuqian Yuan, Kehan Li, Hangjie Yuan, Chaohui Yu, Yuming Jiang, Jiayan Guo, Xin Li, Hao Luo, et al. Rynnvla-002: A unified vision-language-action and world model. *arXiv preprint arXiv:2511.17502*, 2025a.

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*, 2025b.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024.

Hao Chen, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Renrui Zhang, Xiaoqi Li, Xiao He, Yandong Guo, Chi-Wing Fu, Shanghang Zhang, et al. Fast-in-slow: A dual-system foundation model unifying fast manipulation within slow reasoning. *arXiv preprint arXiv:2506.01953*, 2025a.

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*, 2025b.

Xiaowei Chi, Peidong Jia, Chun-Kai Fan, Xiaozhu Ju, Weishi Mi, Kevin Zhang, Zhiyuan Qin, Wanxin Tian, Kuangzhi Ge, Hao Li, et al. Wow: Towards a world omniscient world model through embodied interaction. *arXiv preprint arXiv:2509.22642*, 2025.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Yao Feng, Hengkai Tan, Xinyi Mao, Chendong Xiang, Guodong Liu, Shuhe Huang, Hang Su, and Jun Zhu. Vidar: Embodied video diffusion model for generalist manipulation. *arXiv preprint arXiv:2507.12898*, 2025.

Chenyang Gu, Jiaming Liu, Hao Chen, Runzhong Huang, Qingpo Wuwu, Zhuoyang Liu, Xiaoqi Li, Ying Li, Renrui Zhang, Peng Jia, et al. Manualvla: A unified vla model for chain-of-thought manual generation and robotic manipulation. *arXiv preprint arXiv:2512.02013*, 2025.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations, 2025. URL https://arxiv.org/abs/2412.14803.

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

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loic Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. Dreamgen: Unlocking generalization in robot learning through video world models, 2025. URL https://arxiv.org/abs/2505.12705.

Yueru Jia, Jiaming Liu, Shengbang Liu, Rui Zhou, Wanhe Yu, Yuyang Yan, Xiaowei Chi, Yandong Guo, Boxin Shi, and Shanghang Zhang. Video2act: A dual-system video diffusion policy with robotic spatio-motional modeling, 2026. URL https://arxiv.org/abs/2512.03044.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on Machine Learning*, 2024.

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*, 2026.

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.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024a.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model, 2025. URL https://arxiv.org/abs/2503.00200.

Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18061–18070, 2024b.

Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies, 2025. URL https://arxiv.org/abs/2508.00795.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, Liliang Chen, Shuicheng Yan, Maoqing Yao, and Guanghui Ren. Genie envisioner: A unified world foundation platform for robotic manipulation, 2025. URL https://arxiv.org/abs/2508.05635.

Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. Onetwovla: A unified vision-language-action model with adaptive reasoning. *arXiv preprint arXiv:2505.11917*, 2025.

Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025.

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. *arXiv preprint arXiv:2410.07864*, 2024.

Zhuoyang Liu, Jiaming Liu, Hao Chen, Jiale Yu, Ziyu Guo, Chengkai Hou, Chenyang Gu, Xiangju Mi, Renrui Zhang, Kun Wu, et al. Last _{0}: Latent spatio-temporal chain-of-thought for robotic vision-language-action model. *arXiv preprint arXiv:2601.05248*, 2026.

Weishi Mi, Yong Bao, Xiaowei Chi, Xiaozhu Ju, Zhiyuan Qin, Kuangzhi Ge, Kai Tang, Peidong Jia, Shanghang Zhang, and Jian Tang. Tc-idm: Grounding video generation for executable zero-shot robot motion. *arXiv preprint arXiv:2601.18323*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

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.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pp. 785–799. PMLR, 2023.

Hengkai Tan, Yao Feng, Xinyi Mao, Shuhe Huang, Guodong Liu, Zhongkai Hao, Hang Su, and Jun Zhu. Anypos: Automated task-agnostic actions for bimanual manipulation. *arXiv preprint arXiv:2507.12768*, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Junjie Wen, Minjie Zhu, Yichen Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Chengmeng Li, Xiaoyu Liu, Yaxin Peng, Chaomin Shen, et al. Diffusion-vla: Scaling robot foundation models via unified diffusion and autoregression. *arXiv preprint arXiv:2412.03293*, 2024.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025.

John Won, Kyungmin Lee, Huiwon Jang, Dongyoung Kim, and Jinwoo Shin. Dual-stream diffusion for world-model augmented vision-language-action model. *arXiv preprint arXiv:2510.27607*, 2025.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023.

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

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination?, 2026. URL https://arxiv.org/abs/2603.16666.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025.

Pengfei Zhou, Liliang Chen, Shengcong Chen, Di Chen, Wenzhi Zhao, Rongjun Jin, Guanghui Ren, and Jianlan Luo. Act2goal: From world model to general goal-conditioned policy. *arXiv preprint arXiv:2512.23541*, 2025.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets, 2025. URL https://arxiv.org/abs/2504.02792.

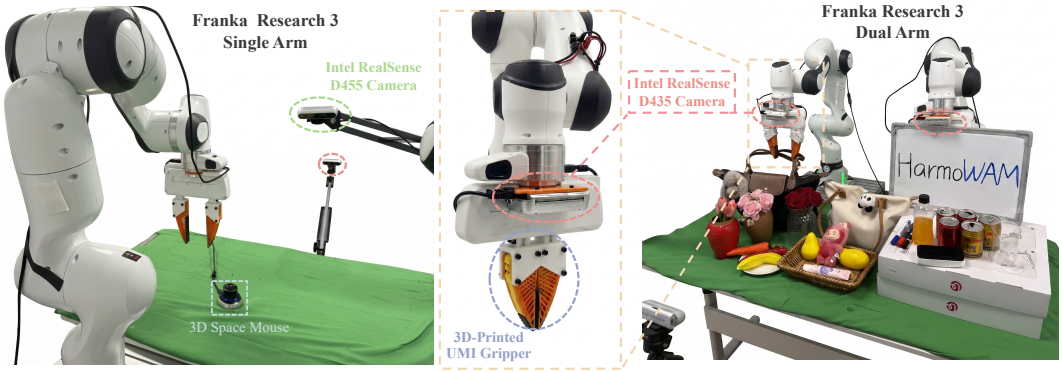
Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *7th Annual Conference on Robot Learning*, 2023.

A Real-World Set-up

Single-Arm Configuration. As shown in Figure 7, our single-arm platform is built on a 7-DoF Franka Research 3 (FR3) manipulator equipped with a 3D-printed UMI gripper for contact-rich precision manipulation tasks. The perception system employs three Intel RealSense cameras to provide multi-view visual feedback: a stationary front-view D435 positioned at approximately $45°$ above the workspace captures a global view, a top-down D455 mounted directly above the manipulation surface provides precise spatial localization, and a wrist-mounted D435 attached to the end-effector supplies close-range observations. All cameras operate at 640×480 resolution.

Dual-Arm Configuration. The dual-arm platform extends to two parallel FR3 arms, each equipped with the same UMI gripper. The camera layout includes one stationary global camera and one wrist-mounted camera per arm. Thus, each dual-arm observation also contains three RGB images, but with two wrist views and one global view. The proprioceptive state is extended to 14 dimensions, with 7 dimensions per arm.

Control and Data Collection. The end-effector poses, gripper states, and other proprioceptive signals of each arm are synchronized with camera observations through a real-time communication interface. All demonstrations are collected via teleoperation using a SpaceMouse 3D controller, with the operator receiving real-time feedback from all camera views.



(a) Single-arm setup of Franka robot. (b) Dual-arm setup of Franka robot.

Figure 7: Real-world robot setup and experimental assets.

B Self-Collected Data

Single-Arm Tasks

1. Pick Fruit to Plate. The robot sequentially picks up a banana and a carrot and places them onto a plate. S_1: grasp and place banana; S_2: grasp and place carrot. The average length is approximately 280 steps and is sequentially evaluated. A stage is considered successful if the robot successfully grasps the corresponding object and places it stably on the plate without dropping it.

2. Stack Coke Cans. The robot stacks three cans one by one, demanding highly precise spatial alignment. S_1: place the second can beside the first; S_2: place the third can on top. The average length is approximately 290 steps and is sequentially evaluated. S_1 is considered successful if the robot places the second can beside the first can with stable contact and without knocking over or displacing the first can. S_2 is considered successful if the robot places the third can on top of the existing cans with stable stacking and without collapsing the stack.

3. Pour Coke into Beaker. The robot grasps a bottle and pours its contents into a beaker, testing fine-grained rotational control. S_1: grasp bottle; S_2: pour into beaker. The average length is approximately 310 steps and is sequentially evaluated. S_1 is considered successful if the robot securely grasps and lifts the bottle, and S_2 is considered successful if the robot tilts the bottle toward the beaker and pours the contents into the beaker without spilling outside the target region.

4. Write "Yes". The robot picks up a marker and writes "Y", "e", "s" on a whiteboard in sequence. S_1: write "Y"; S_2: write "e"; S_3: write "s". The average length is approximately 310 steps and is sequentially evaluated. A stage is considered successful if the robot writes the corresponding character legibly on the whiteboard.

Dual-Arm Tasks

1. Put Flowers in Vase. The left arm picks a flower and hands it to the right arm, which inserts it into a vase, requiring precise bimanual coordination and tight-tolerance insertion. S_1: pick flower; S_2: bimanual handover; S_3: insert into vase. The average length is approximately 280 steps and is sequentially evaluated. S_1 is considered successful if the left arm securely grasps and lifts the flower without dropping it. S_2 is considered successful if the flower is successfully transferred from the left arm to the right arm while maintaining a stable grasp and without dropping the flower. S_3 is considered successful if the right arm inserts the flower into the vase.

2. Put Items to Bag and Zip. Both arms collaborate to place items into a bag and zip it closed, which is the longest-horizon task. $S_1 \rightarrow S_2$: pick up item and place into bag; $S_3 \rightarrow S_4 \rightarrow S_5$: one arm grips the bag to hold it steady, the other grips and pulls the zipper to close. The average length is approximately 400 steps and is sequentially evaluated. S_1 is considered successful if the robot securely grasps and lifts the target item without dropping it. S_2 is considered successful if the item is placed inside the bag. S_3 is considered successful if one arm firmly holds the bag in place without significantly deforming or displacing it. S_4 is considered successful if the other arm successfully grasps the zipper puller. S_5 is considered successful if the zipper is pulled along the closing direction and the bag is closed.

C Real-World Experiment Details.

C.1 Baselines.

$\pi_{0.5}$: (Intelligence et al., 2025) A PaliGemma-based VLA with a hierarchical architecture that generates continuous actions via flow matching. **QwenVLA-OFT:** (Bai et al., 2025) An implementation that adopts Qwen3-VL as the backbone and performs parallel action prediction with a ℓ_1 regression objective. **VPP:** (Hu et al., 2025) A video-prediction-based policy that extracts intermediate features from a video diffusion model to condition a separate diffusion policy head. **Cosmos-Policy:** (Kim et al., 2026) A joint video-action modeling baseline that encodes actions as latent-space frames and denoises them together with video frames in the Cosmos video model. **Wan+AnyPos:** (Wan et al., 2025; Tan et al., 2025) An Imagine-then-Execute baseline where Wan2.2-TI2V-5B predicts future trajectory videos and AnyPos serves as an inverse dynamics model, inferring the corresponding action from each predicted frame.

For all baselines, we follow the training protocols recommended in their original papers whenever applicable. $\pi_{0.5}$, Cosmos-Policy, and Wan2.2-TI2V-5B are initialized from their officially released pretrained checkpoints and then fine-tuned using the configurations and hyperparameters suggested by the corresponding original papers. For AnyPos, we train the model from scratch on the downstream dataset following the recommended settings in the original paper. For VPP, we fine-tune Stable Video Diffusion and replace its action head with the same 1B-parameter Action Diffusion Transformer used in our method to ensure a fair comparison. QwenVLA-OFT is built upon Qwen3-VL-4B. Following prior work (Liu et al., 2026), we additionally pretrain the model on over 400K cross-embodiment trajectories to endow it with robotic priors, and further fine-tune it for robotic control using a parallelized ℓ_1 regression objective. All methods are trained on the same real-world demonstration dataset, given the same task instructions, and evaluated using the same multi-view RGB observations, except for Wan+AnyPos, which uses only the front-view RGB observation. All methods predict actions in the same control space: 7 dimensions for single-arm tasks and 14 dimensions for dual-arm tasks. We evaluate all methods under the same real-world testing protocol.

C.2 Training Protocol.

All models are trained on 8 NVIDIA H20 GPUs. In the first stage, the world model is initialized from the official Wan2.2-TI2V-5B checkpoint (Wan et al., 2025) and fully fine-tuned on our real-world demonstration videos using a conditional Flow Matching objective. It takes the current

front-view RGB observation at 256×320 resolution and predicts 13-frame videos. In the second stage, the world model is frozen to provide explicit video predictions and implicit latent features for training the two action experts. The latent features are extracted from the final DiT block at the last denoising step, yielding 80 spatial tokens with 3,072-dimensional features per latent timestep. The predictive expert uses RGB observations from all three camera views defined in Appendix A. Each image is padded to a square and processed at 384×384 resolution by SigLIP, with the resulting patch features concatenated along the token dimension. The reactive expert uses the predicted front-view future frames and their corresponding latent features. Each image input to DINOv2 is resized to 518×518 and normalized. Removing the CLS and register tokens yields 1,369 patch tokens with 768-dimensional features per image. The corresponding Wan features are pooled along the channel dimension to 768 dimensions and concatenated with the DINOv2 tokens, followed by adaptive pooling to 1,369 tokens. The gating module uses only the current front-view RGB observation, encoded by the frozen Wan VAE encoder, and only its MLP is trained (Appendix D.3). The predictive expert is trained with a diffusion denoising loss, the reactive expert with a Smooth L1 loss, and the gating network with a binary cross-entropy loss. Specifically, the loss of the reactive expert is defined as:

$$\mathcal{L}_{\text{react}} = \mathbb{E}\left[d(\hat{\mathbf{a}}_{t+1:t+H}, \mathbf{a}_{t+1:t+H})\right]. \tag{5}$$

where $d(\cdot, \cdot)$ denotes the Smooth L1 distance:

$$d(x, \hat{x}) = \begin{cases} 0.5 \cdot \frac{(x-\hat{x})^2}{\beta}, & \text{if } |x - \hat{x}| < \beta, \\ |x - \hat{x}| - 0.5\beta, & \text{otherwise,} \end{cases} \tag{6}$$

where β is set to 0.1. The overall objective is a weighted sum of all sub-losses. The action space is 7-dimensional for single-arm tasks and 14-dimensional for dual-arm tasks.

Inference Setup. Real-robot inference runs on a single NVIDIA RTX 4090 GPU. The world model is invoked once per action chunk, using five denoising steps to generate a 13-frame video and its latent features. The gate selects one action expert to generate a 12-step action chunk. The model inference frequency is approximately 4 Hz, corresponding to an action-output throughput of 48 predicted actions/s.

C.3 Out-of-Domain (OOD) Scenario Construction

Unseen Background. To evaluate robustness against visual context shifts, we construct background OOD scenes by introducing unseen distractor objects and lighting changes. Specifically, 5–8 objects that never appear during training are randomly placed in the workspace, with variations in color, texture, and geometry. These distractors are distributed near the target object and in peripheral tabletop regions to increase scene clutter. We further modify ambient lighting by changing light intensity and incident direction, producing appearance changes, shadows, and local reflections not observed during training. This setting tests the model's ability to maintain reliable perception and stable manipulation in cluttered and visually perturbed environments.

Unseen Position. Target objects are placed in workspace regions outside the spatial coverage of training demonstration trajectories. Specifically, we partition the manipulation surface into training and test regions, where the test region is spatially disjoint from the regions covered by object positions during data collection. This setting evaluates the model's generalization capability under unseen spatial coordinates.

Unseen Objects. Target objects in each task are replaced with substitutes that differ significantly from the training objects in visual appearance and geometry. For example, a carrot is replaced with a pepper, a standard Coke can is replaced with a Red Bull can with different body dimensions and shape, and the original flower is replaced with flowers of different colors and varying numbers of blossoms. This setting evaluates the model's robustness to object-level semantic and geometric variations, particularly when grasping strategies and interaction patterns need to be adaptively adjusted.

D Additional Method Details

D.1 Additional Preliminaries

IDM-based Action Inference. The world model predicts future visual frames and spatio-temporal latent representations, which are not directly executable by the robot. To translate these predictions

into low-level control actions, some works (Tan et al., 2025; Mi et al., 2026; Feng et al., 2025) adopt an Inverse Dynamics Model (IDM), denoted by π_{IDM}. Given a future video trajectory $\mathbf{V}_{t:t+H}$, the IDM produces the action sequence: $\mathbf{a}_{t+1:t+H} \sim \pi_{\text{IDM}}(\cdot \mid \mathbf{V}_{t+1:t+H})$. The output follows the same action paradigm as defined above.

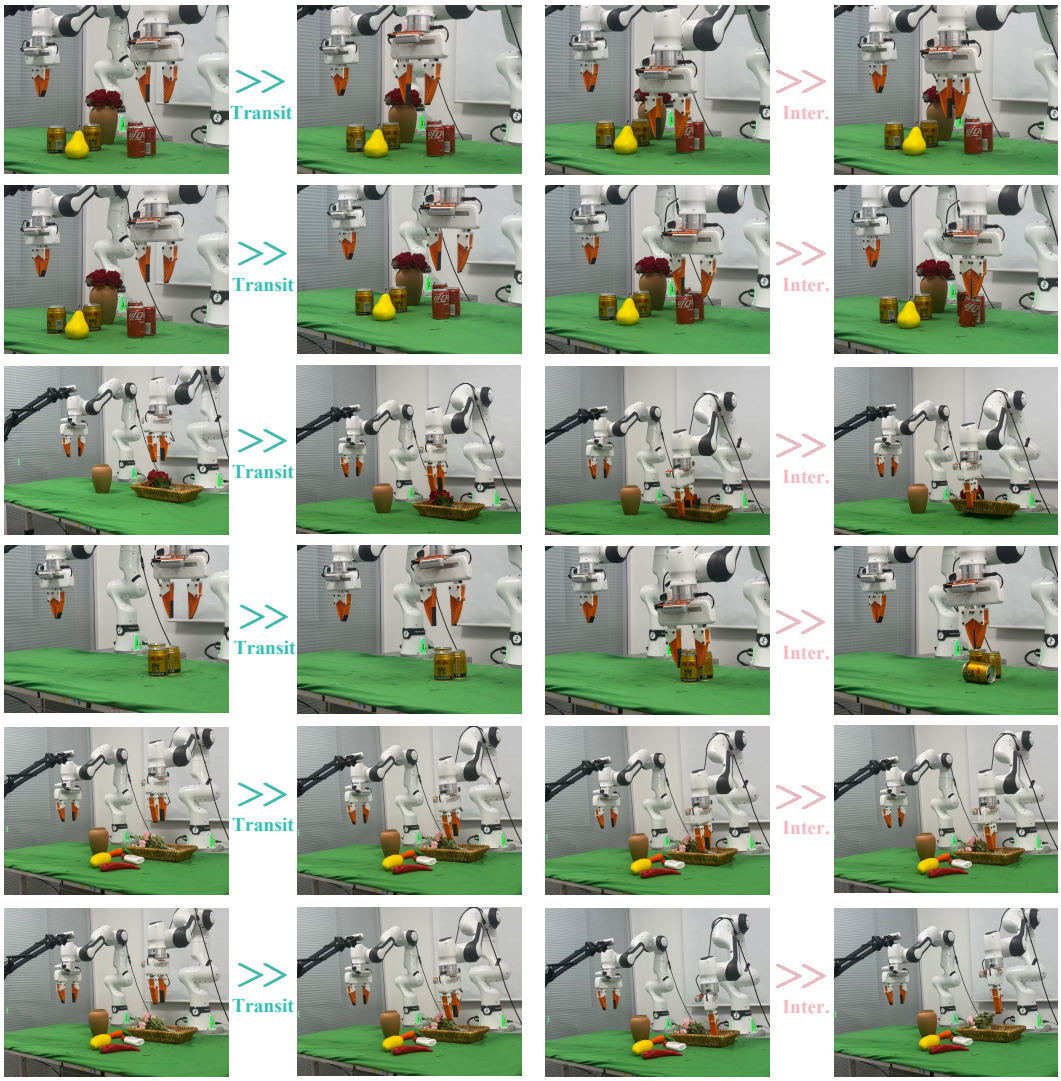


Figure 8: Representative failure cases of the two World Action Models paradigms under OOD scenarios. The first three rows show typical failures of the *Joint Modeling* baseline and the last three rows show common failures of the *Imagine-then-Execute* baseline.

D.2 Motivation Experiments Details

We conduct the motivation study on two representative real-world manipulation tasks: *Put Flowers in Vase*, which requires long-horizon dual-arm coordination, and *Stack Coke Cans*, which demands precise, fine-grained stacking manipulation, under both in-domain (ID) settings and three out-of-domain (OOD) scenarios, including background, position, and semantic variations, with detailed OOD settings provided in Appendix C.3 and OOD visualizations shown in Appendix E.2. For each task and domain setting, we evaluate both the *Transit* and *Interaction* phases over 10 real-world trials. *Transit* measures the ability to reach the target vicinity from the initial position or move between objects, reflecting generalization. *Interaction* measures the success of core actions such as picking, placing, stacking, handover, or insertion, reflecting precision. To ensure a controlled comparison, both baselines use the same Wan2.2-TI2V-5B checkpoint, downstream demonstrations, ac-

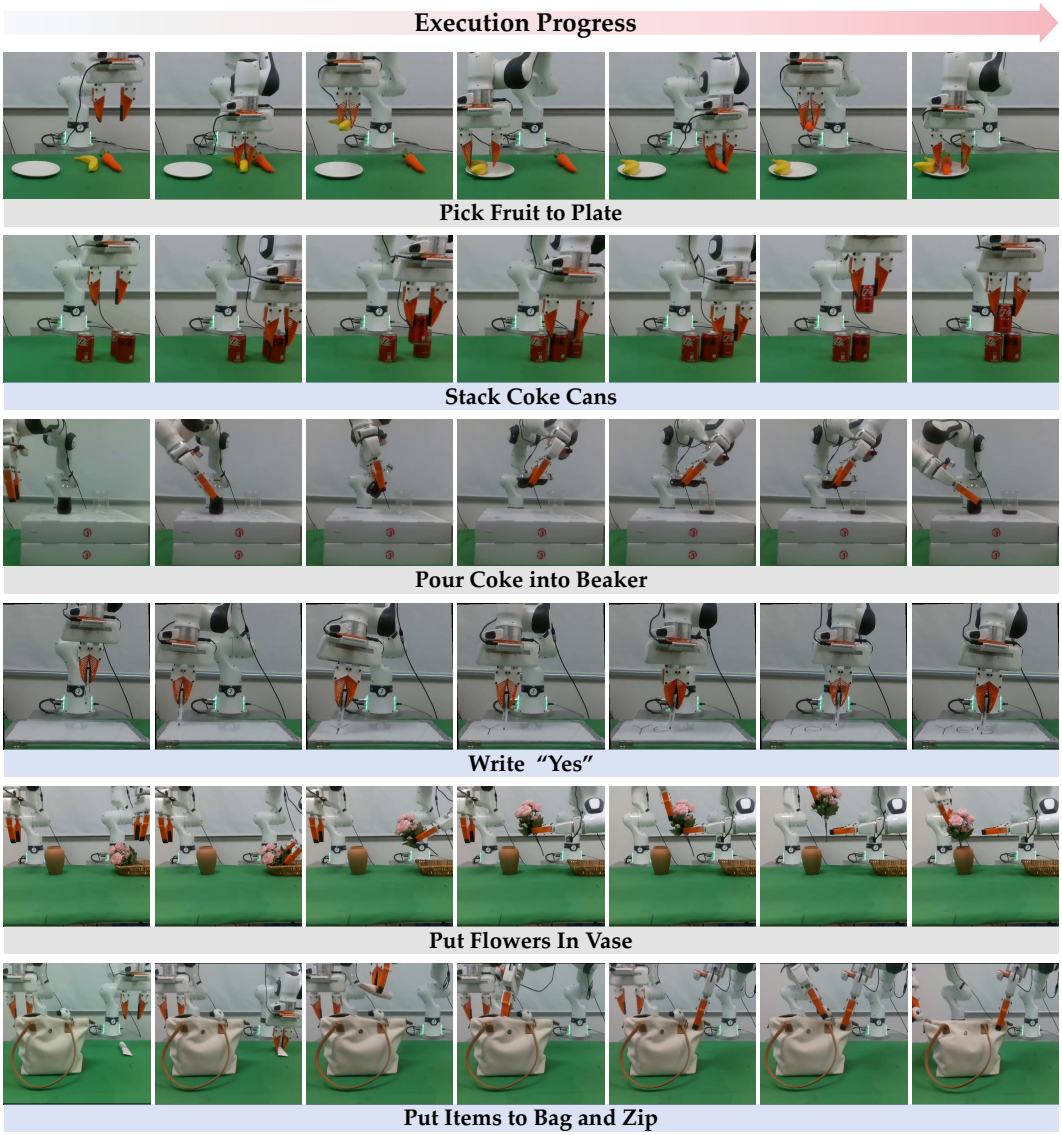


Figure 9: Visualization of complete execution sequences on six real-world manipulation tasks.

tion space, robot platform, and ID/OOD evaluation protocol. Their action-generation architectures remain paradigm-specific by design: the *Imagine-then-Execute* baseline first predicts explicit future observations and then infers actions through an inverse-dynamics pathway, whereas the *Joint Modeling* baseline directly conditions action generation on world-model latent features. These differences arise from the intrinsic world-to-action formulations of the two paradigms, rather than from additional uncontrolled factors. For the *Joint Modeling* baseline, interaction performance is evaluated by initializing the robot near the target object. For the *Imagine-then-Execute* baseline, no such reinitialization is required, as it autonomously reaches the target vicinity during execution. This protocol decouples interaction precision from transit failures and enables a direct comparison of the manipulation capabilities of the two paradigms after the target vicinity is reached. Figure 8 visualizes representative failure cases of the two paradigms. The first three rows show typical failures of the *Joint Modeling* baseline. Under OOD scenarios, changes in background appearance or object position can shift the predicted trajectory away from the target region. As a result, the end-effector fails to move above the manipulated object or stops at an incorrect spatial location, causing failure in the transit phase before meaningful object contact is established. The last three rows show common failures of the *Imagine-then-Execute* baseline. Although it can often generate a reasonable transit trajectory and move the gripper to the vicinity of the target object, the inferred actions may still be inaccurate during the interaction phase. For example, the gripper may close at an offset grasp point, or open or close prematurely, leading to unsuccessful grasping, placement, or stacking.

D.3 Process-Adaptive Gating Details

The Process-Adaptive Gating Mechanism selects between the predictive and reactive action experts according to the current stage of task execution. Following the event-based labeling procedure of Perceiver-Actor (Shridhar et al., 2023), we automatically generate frame-level transit/interaction labels from proprioceptive signals recorded in demonstrations, without manual keyframe annotation or frame-by-frame stage labeling. The pipeline primarily detects key interaction events from gripper-state transitions and labels the surrounding ± 20 frames as interaction to cover grasping, release, and related interaction motions, while labeling the remaining frames as transit. For example, in *Put Items to Bag and Zip*, labels are generated entirely from gripper-state transitions, without height thresholds or geometric rules. These transitions include gripper opening or closing when grasping or releasing objects, holding the bag, or operating the zipper. Only in rare cases where key interactions do not involve gripper-state transitions do we select auxiliary signals, such as height thresholds, after which labels are still generated automatically. The same rule can generally be reused for common tasks with minimal additional manual intervention. For dual-arm tasks, a timestep and its surrounding temporal window are automatically labeled as interaction whenever either arm satisfies the event criterion. These proprioceptive signals are used only for offline label construction. During inference, the gate receives only features of the current observation extracted by the frozen Wan VAE encoder and maps them to a routing score using a lightweight MLP.

We evaluate the frozen gating module offline on held-out demonstrations of *Put Flowers in Vase* excluded from gate training. Under the ID setting, it achieves an average frame-level accuracy of 96.95% on 1,637 test frames from six trajectories. Under the Background, Position, and Object OOD settings, we evaluate five trajectories per setting, totaling 15 trajectories and 4,207 test frames, and obtain an average frame-level accuracy of 94.77%. The consistently high stage-recognition accuracy under both ID and OOD settings supports the gate's robustness to changes in background, object position, and object identity.

We further analyze gating behavior during real-robot closed-loop execution of *Put Flowers in Vase*, using one successful rollout from each of the Background, Position, and Object OOD settings, together with one representative failure rollout. During execution, the gate outputs a score s_t from the current visual observation for each 12-step action chunk and selects an action expert using a threshold of 0.5. The robot then executes the selected expert's actions. Table 4 summarizes the gate scores in these rollouts. In successful rollouts, scores remain low during initial and inter-stage transit and rise during grasping, bimanual handover, and insertion into the vase. This consistent low–high–low–high pattern across OOD settings indicates that gating decisions align with task-stage transitions during actual execution, supporting the gate's closed-loop robustness. The failure rollout exhibits a similar overall pattern, but its shorter first interaction-routing interval may result in insufficient intervention by the predictive action expert, potentially contributing to execution deviations.

Table 4: Closed-loop gate-score statistics on *Put Flowers in Vase*. Consecutive action chunks with the same routing decision are grouped using a threshold of 0.5. Entries report the mean [minimum–maximum] score, with action-chunk indices in parentheses. Single-chunk intervals report only their score.

Rollout	Result	Initial Transit	First Interaction	Inter-stage Transit	Later Interaction
Background OOD	Success	0.37 (1)	0.87 [0.56–0.99] (2–9)	0.10 [0.02–0.22] (10–14)	0.92 [0.75–1.00] (15–24)
Position OOD	Success	0.32 (1)	0.80 [0.52–0.97] (2–5)	0.30 [0.15–0.41] (6–10)	0.97 [0.80–1.00] (11–19)
Object OOD	Success	0.12 [0.06–0.18] (1–2)	0.92 [0.84–0.98] (3–6)	0.10 [0.03–0.19] (7–13)	0.88 [0.59–1.00] (14–20)
Representative Failure	Failure	0.27 [0.11–0.47] (1–3)	0.66 [0.63–0.70] (4–5)	0.34 [0.23–0.43] (6–9)	0.94 [0.57–1.00] (10–20)

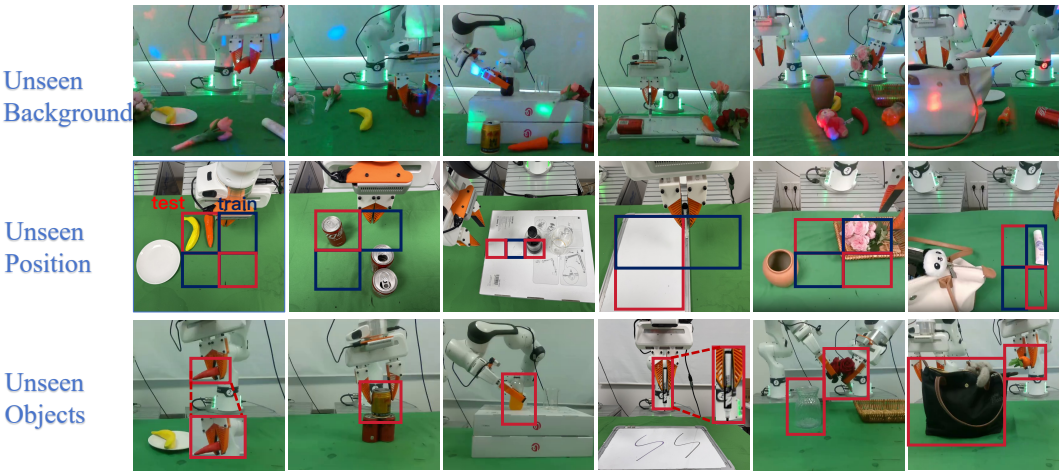


Figure 10: Visualization of the complete examples under three OOD settings: unseen background, unseen position, and unseen objects.

Window Sensitivity Protocol. For Figure 6(d), we keep the world model and both action experts fixed and retrain only the gate with interaction-label windows of ± 10, ± 20 (default), and ± 30 frames. We evaluate each gate on *Put Flowers in Vase* over 10 real-robot trials under each of the ID, Background, Position, and Object settings, using the success-rate metric in Section 4.1. The average success rate is the mean across these four settings.

E Additional Visualization

E.1 Additional Task Execution Visualizations.

Figure 9 illustrates the complete execution sequences of HarmoWAM across all six tasks. We observe that HarmoWAM produces smooth and continuous motion trajectories, particularly in phases requiring fine-grained control, such as spatial alignment during can stacking, sustained angular control during pouring, and tight-tolerance alignment during flower insertion.

E.2 Additional Generalization Visualizations.

Figure 10 shows the complete examples under all three OOD dimensions: background, position, and semantic variations. Scenario construction details are provided in Appendix C.3.

F Additional Quantitative and Qualitative Results

F.1 In-Domain(ID) Stage-Wise Results.

Tables 5 and 6 report the complete cumulative stage success rates on in-domain (ID) tasks, providing a fine-grained view of subgoal performance beyond task-level averages. For a task with K stages and N trials, let n_k count trials completing all stages through k. The cumulative stage success rate is

Table 5: Cumulative stage success rates for ID single-arm manipulation tasks. The final-stage rate equals the end-to-end success rate. Avg. averages all listed stage entries.

Method	Pick Fruit to Plate		Stack Coke Cans		Pour Coke into Beaker		Write "Yes"			Avg
	S1 → S2		S1 → S2		S1 → S2		S1 → S2 → S3			
$\pi_{0.5}$	0.85	0.75	0.75	0.60	0.80	0.70	0.90	0.80	0.80	0.77
VPP	0.85	0.75	0.65	0.55	0.80	0.75	0.85	0.75	0.60	0.73
Wan+AnyPos	0.90	0.85	0.85	0.35	**0.90**	0.65	0.90	0.70	0.55	0.74
QwenVLA-OFT	0.80	0.75	0.35	0.25	0.75	0.70	0.80	0.75	0.60	0.64
Cosmos-Policy	**0.95**	0.90	0.75	0.55	0.85	0.75	**0.95**	0.85	0.70	0.81
Ours	**0.95**	**0.95**	**0.95**	**0.85**	**0.90**	**0.85**	**0.95**	**0.90**	**0.90**	**0.91**

Table 6: Cumulative stage success rates for ID dual-arm manipulation tasks. The final-stage rate equals the end-to-end success rate. Avg. averages all listed stage entries.

Method	Put Flowers in Vase			Put Items to Bag and Zip					Avg
	S1 → S2 → S3			S1 → S2 → S3 → S4 → S5					
$\pi_{0.5}$	**0.95**	0.80	0.40	0.85	0.75	0.70	0.55	0.50	0.69
Wan+AnyPos	0.85	0.60	0.15	0.75	0.60	0.45	0.40	0.40	0.53
Cosmos-Policy	**0.95**	**0.95**	0.35	**0.90**	0.80	0.70	0.65	0.55	0.73
Ours	**0.95**	0.90	**0.70**	**0.90**	**0.90**	**0.85**	**0.80**	**0.80**	**0.85**

$p_k = n_k/N$, and the average cumulative stage success rate is $\bar{p} = K^{-1}\sum_{k=1}^{K} p_k$. Since execution is continuous without stage reinitialization, the end-to-end success rate equals p_K. For long-horizon tasks, this metric complements the end-to-end success rate by capturing execution progress and distinguishing early failures from those near task completion. For single-arm tasks, HarmoWAM consistently achieves high success across different subgoals. In *Stack Coke Cans*, which requires precise spatial alignment, HarmoWAM obtains 95% success in the first stage and 85% in the second stage, demonstrating reliable end-effector control across repeated stacking operations. In *Pour Coke into Beaker*, which emphasizes continuous pose control, HarmoWAM achieves 90% in the grasping stage and 85% in the pouring stage, indicating that it can effectively maintain the relative position and orientation between the bottle and the beaker. Across all single-arm tasks, HarmoWAM reaches a success rate of 91%, outperforming the next-best baseline, Cosmos-Policy, by 10 percentage points. For dual-arm tasks, the stage-wise results further highlight HarmoWAM's advantage in complex collaborative procedures. In *Put Flowers in Vase*, most baselines can complete early picking or partial handover, but struggle with the final insertion stage. For example, Wan+AnyPos drops from 60% in the handover stage to 15% in the insertion stage, while HarmoWAM maintains 70% success in the final stage. This shows that HarmoWAM better preserves precise control after bimanual handover. In *Put Items to Bag and Zip*, which involves multiple consecutive subgoals such as item placement, bag stabilization, zipper grasping, and zipper pulling, baseline performance generally decreases in later stages. In contrast, HarmoWAM maintains 85%, 80%, and 80% success across the final three stages, indicating stronger action coherence over long horizons and reduced error propagation. Overall, HarmoWAM achieves a success rate of 85% on dual-arm tasks, outperforming the next-best baseline, Cosmos-Policy, by 12 percentage points. Overall, the ID stage-wise results further show that the advantage of HarmoWAM is not limited to higher task-level averages. It is also reflected in key subgoals such as stacking alignment, continuous pouring, insertion after bimanual handover, and zipper closure. These steps require finer local control than ordinary pick-and-place operations and are where baseline methods most frequently fail. These results highlight HarmoWAM's advantage in using a world model to provide spatio-temporal conditions while

Table 7: Cumulative stage success rates for OOD single-arm manipulation tasks. The final-stage rate equals the end-to-end success rate. Avg. averages all listed stage entries.

Method	Gen Scenario	Pick Fruit		Stack Cans		Pour Coke		Write "Yes"			Avg
		S1 → S2		S1 → S2		S1 → S2		S1 → S2 → S3			
$\pi_{0.5}$	Background	0.75	0.55	0.45	0.35	0.70	0.55	0.80	0.65	0.55	0.59
	Position	0.25	0.15	0.10	0.05	0.45	0.40	0.60	0.55	0.50	0.34
	Objects	0.65	0.45	0.50	0.35	0.55	0.50	0.85	0.60	0.45	0.54
VPP	Background	0.50	0.25	0.35	0.15	0.70	0.65	0.55	0.40	0.25	0.42
	Position	0.30	0.25	0.15	0.05	0.25	0.15	0.50	0.35	0.15	0.24
	Objects	0.80	0.55	0.55	0.25	0.70	0.65	0.70	0.45	0.35	0.56
Wan+AnyPos	Background	0.60	0.55	0.60	0.25	0.80	0.50	0.80	0.60	0.55	0.58
	Position	0.85	0.80	0.55	0.25	0.35	0.30	0.75	0.40	0.35	0.51
	Objects	0.80	0.45	0.70	0.60	0.75	0.60	0.80	0.70	0.55	0.66
QwenVLA-OFT	Background	0.70	0.25	0.15	0.10	0.60	0.55	0.70	0.60	0.60	0.47
	Position	0.45	0.10	0.15	0.05	0.40	0.25	0.55	0.40	0.30	0.29
	Objects	0.65	0.60	0.15	0.05	0.75	0.70	0.60	0.50	0.45	0.49
Cosmos-Policy	Background	**0.90**	0.80	0.60	0.15	0.80	0.65	0.85	0.45	0.40	0.62
	Position	0.25	0.15	0.30	0.10	0.15	0.15	0.65	0.50	0.50	0.31
	Objects	0.85	0.75	0.60	0.15	0.70	0.65	0.35	0.15	0.15	0.48
Ours	Background	**0.90**	**0.90**	**0.90**	**0.65**	**0.85**	**0.80**	**0.90**	**0.85**	**0.70**	**0.83**
	Position	**0.95**	**0.90**	**0.85**	**0.70**	**0.85**	**0.75**	**0.85**	**0.85**	**0.80**	**0.83**
	Objects	**0.95**	**0.80**	**0.95**	**0.75**	**0.90**	**0.85**	**0.90**	**0.85**	**0.85**	**0.87**

the Process-Adaptive Gating coordinates the predictive action expert and the reactive action expert, enabling the framework to achieve reliable and accurate execution across complex tasks.

F.2 Out-of-Domain(OOD) Stage-Wise Results.

Tables 7 and 8 present the OOD cumulative stage success rates, following the definitions in Appendix F.1. The results show that different distribution shifts challenge different aspects of robot execution: background variation mainly affects visual grounding under distractors and illumination changes, position variation tests whether policies can act beyond the spatial distribution of the supervised training data, and semantic variation stresses the model's understanding of object geometry, functionality, and contact relations. Under background variation, HarmoWAM achieves a success rate of 83% on single-arm tasks and 76% on dual-arm tasks, outperforming the strongest baselines by 21 and 15 percentage points, respectively. Several baselines show clear degradation in grasping or placement stages under cluttered scenes. For example, in *Stack Coke Cans*, the second-stage success of all baselines remains no higher than 35%, while HarmoWAM reaches 65%. In dual-arm tasks, HarmoWAM also maintains 60% success in the final insertion stage of *Put Flowers in Vase*, compared with 30% for the best baseline. These results indicate that HarmoWAM remains robust by using the world model's predictive and generalization capabilities to forecast task-relevant visual dynamics while suppressing distractors, enabling downstream action pathways to focus on object interactions and reliably execute actions in cluttered environments. Under position variation, the performance gap becomes more pronounced. HarmoWAM achieves 83% on single-arm tasks and 70% on dual-arm tasks, while the best baselines only reach 51% and 45%, respectively. This suggests that baseline models are strongly constrained by the spatial coverage of the supervised fine-tuning data. In *Stack Coke Cans*, HarmoWAM achieves 85% and 70% across the two stages, whereas the best baseline drops to 55% and 25%. In *Put Flowers in Vase*, HarmoWAM maintains 90%, 85%, and 65% across the three stages, substantially outperforming Wan+AnyPos, which drops

Table 8: Cumulative stage success rates for OOD dual-arm manipulation tasks. The final-stage rate equals the end-to-end success rate. Avg. averages all listed stage entries.

Method	Gen Scenario	Put Flowers in Vase			Put Items to Bag and Zip					Avg
		S1 → S2 → S3			S1 → S2 → S3 → S4 → S5					
$\pi_{0.5}$	Background	**0.90**	0.65	0.30	0.70	0.65	0.65	0.50	0.50	0.61
	Position	0.50	0.35	0.25	0.35	0.30	0.25	0.20	0.20	0.30
	Objects	0.80	0.65	0.35	0.65	0.55	0.45	0.40	0.40	0.53
Wan+AnyPos	Background	0.75	0.55	0.10	0.45	0.40	0.40	0.35	0.30	0.41
	Position	0.70	0.50	0.15	0.60	0.55	0.40	0.35	0.35	0.45
	Objects	0.65	0.45	0.10	0.50	0.45	0.45	0.40	0.35	0.42
Cosmos-Policy	Background	0.65	0.45	0.15	0.65	0.50	0.50	0.40	0.35	0.46
	Position	0.35	0.30	0.10	0.35	0.25	0.25	0.10	0.10	0.23
	Objects	0.80	0.65	0.05	0.70	0.55	0.40	0.35	0.20	0.46
Ours	Background	0.85	**0.75**	**0.60**	**0.85**	**0.85**	**0.80**	**0.70**	**0.70**	**0.76**
	Position	**0.90**	**0.85**	**0.65**	**0.70**	**0.65**	**0.65**	**0.60**	**0.60**	**0.70**
	Objects	**0.95**	**0.85**	**0.60**	**0.85**	**0.85**	**0.80**	**0.80**	**0.75**	**0.81**

from 70% to 15% by the final insertion stage. These results show that HarmoWAM leverages the world model's generalization capability to guide the reactive action expert beyond the constraints of the SFT data distribution, while relying on the predictive action expert for fine-grained manipulation at critical interaction stages. Under semantic variation, HarmoWAM achieves the highest success rates, reaching 87% on single-arm tasks and 81% on dual-arm tasks, compared with the strongest baselines at 66% and 53%. The advantage is especially clear in contact-sensitive stages. For example, in *Stack Coke Cans*, HarmoWAM reaches 95% and 75%, while the best baseline reaches 70% and 60%. In *Put Flowers in Vase*, HarmoWAM achieves 95%, 85%, and 60%, whereas the best baseline only reaches 80%, 65%, and 35%. This further demonstrates that our approach can effectively harness the world model's learned semantic priors, enabling the two action experts to be coordinated through explicit and implicit conditioning for reliable object approach and interaction with unseen objects. This demonstrates that HarmoWAM can maintain generalizable transit execution under visual and spatial shifts while ensuring precise interaction during contact-intensive manipulation.

F.3 Additional Ablation

Table 9: Ablation study on the number of denoising steps of the world model. Inference setup is described in Appendix C.2.

Denoising Steps	Success Rate (%)	Model Inference Frequency (Hz)
3	80	4
5	**85**	**4**
10	85	3.6
50	87	3

We ablate the number of denoising steps in the world model on the *Put Flowers in Vase* task under the ID setting. We evaluate 3, 5, 10, and 50 steps over 20 real-robot trials per configuration and compare their success rates and model inference frequencies. This study aims to identify the best trade-off between video prediction quality and generation efficiency. As shown in Table 9, using 3 steps results in a lower success rate (80%) due to coarse and blurry video predictions, while 5 steps

lead to an improved success rate of 85% without reducing inference frequency. Increasing to 10 or 50 steps does not significantly improve success rates (85% and 87%, respectively) but decreases inference frequency, indicating diminishing returns for higher denoising steps. We further visualize the generated videos in Figure 11: videos with 3 steps are noticeably blurred and lack key details, while 5-step videos preserve clarity of crucial action stages and object locations, ensuring reliable observation and task execution.

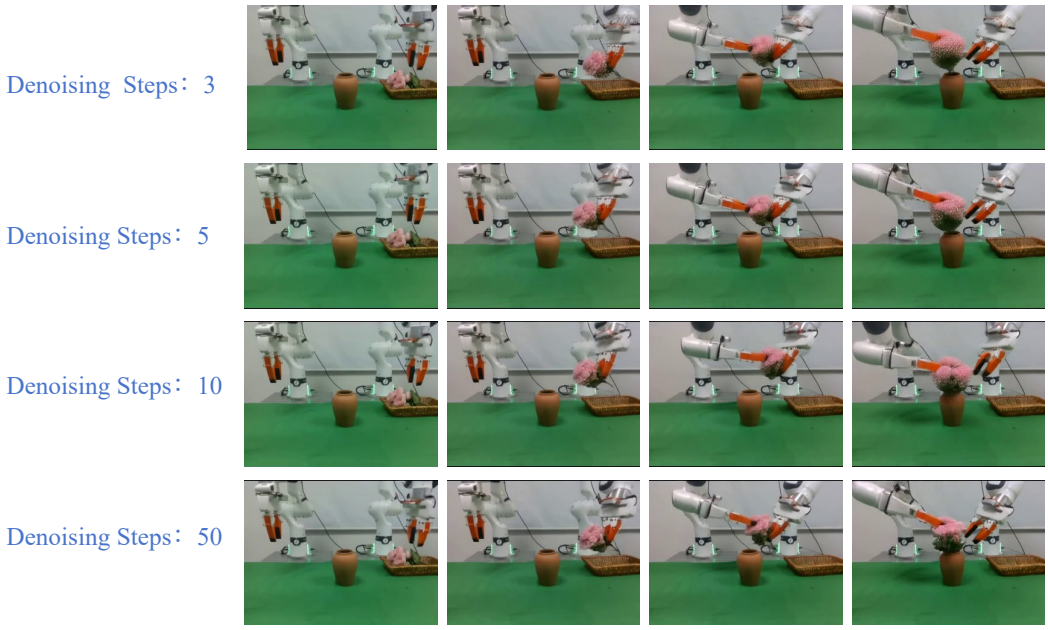


Figure 11: Visual comparison of generated videos under different denoising steps.

F.4 RoboTwin 2.0 simulation results

We additionally evaluate HarmoWAM on the 50 bimanual manipulation tasks in RoboTwin 2.0 (Chen et al., 2025b), built on the SAPIEN simulator with the ALOHA-AgileX dual-arm robot. Method configurations and training procedures follow the real-world experiments in Appendix C.1 and C.2, except that each method is trained using 50 expert demonstrations per task from the Clean setting only. We evaluate each method over 50 rollouts per task with different seeds under the Clean setting. Success is determined by the benchmark's task-completion criteria.

Table 10 reports the complete Clean results for all 50 tasks and all six methods. HarmoWAM achieves the highest average success rate of 67.60%, compared with 56.56% for $\pi_{0.5}$ and 50.00% for Wan+AnyPos. These results further demonstrate HarmoWAM's effectiveness across diverse bimanual manipulation tasks in simulation.

Table 10: Complete results on the 50 RoboTwin 2.0 tasks under the Clean setting. Each method is trained on 50 Clean demonstrations per task and evaluated over 50 Clean rollouts per task.

Task	$\pi_{0.5}$	Cosmos-Policy	Wan+ AnyPos	VPP	QwenVLA-OFT	**HarmoWAM**
Adjust Bottle	84	48	60	28	88	100
Beat Block Hammer	88	16	64	48	44	84
Blocks Ranking RGB	0	4	44	0	0	0
Blocks Ranking Size	20	8	24	0	0	40
Click Alarmclock	64	68	88	52	0	36
Click Bell	56	76	100	32	0	52
Dump Bin Bigbin	96	16	64	48	80	88
Grab Roller	96	24	96	72	88	100
Handover Block	8	12	4	12	0	60
Handover Mic	88	32	56	76	0	100
Hanging Mug	16	12	4	4	0	16
Lift Pot	96	16	20	32	4	84
Move Can Pot	32	12	8	4	32	76
Move Pillbottle Pad	56	16	32	0	44	56
Move Playingcard Away	72	16	56	12	48	88
Move Stapler Pad	24	12	44	0	4	32
Open Laptop	72	72	40	36	80	96
Open Microwave	56	52	4	84	68	12
Pick Diverse Bottles	44	16	76	4	0	80
Pick Dual Bottles	44	28	72	4	0	96
Place A2B Left	60	16	52	0	32	72
Place A2B Right	52	20	28	4	44	72
Place Bread Basket	40	12	48	0	32	68
Place Bread Skillet	56	16	40	0	52	84
Place Burger Fries	80	20	92	20	76	88
Place Can Basket	48	8	68	20	32	36
Place Cans Plasticbox	52	12	84	0	16	96
Place Container Plate	84	64	92	24	68	92
Place Dual Shoes	36	8	0	0	24	72
Place Empty Cup	80	20	76	32	56	96
Place Fan	48	12	52	0	44	52
Place Mouse Pad	36	14	20	0	4	24
Place Object Basket	56	4	60	16	44	72
Place Object Scale	68	12	44	0	16	64
Place Object Stand	80	40	68	0	40	80
Place Phone Stand	56	8	44	0	24	72
Place Shoe	68	16	48	12	36	88
Press Stapler	76	88	16	28	76	76
Put Bottles Dustbin	24	8	16	12	8	84
Put Object Cabinet	48	12	8	4	16	56
Rotate QRcode	68	12	36	40	52	64
Scan Object	48	12	36	0	0	56
Shake Bottle	100	96	100	64	80	96
Shake Bottle Horizontally	100	96	100	56	88	100
Stack Blocks Three	0	4	48	0	0	0
Stack Blocks Two	16	8	100	0	0	44
Stack Bowls Three	72	12	44	32	32	76
Stack Bowls Two	84	24	88	44	68	96
Stamp Seal	40	8	24	0	20	44
Turn Switch	40	28	12	20	28	64
Average	56.56	25.32	50.00	19.52	33.76	**67.60**

G Failure Case Analysis

Through real-world experiments on the Franka platform, we observed three specific failure cases encountered when HarmoWAM controlled the robot, as shown in Figure 12. Red boxes highlight the key errors.

Tilted Stacking. In stacking tasks, HarmoWAM can usually complete the grasping and lifting motions, but the final placement may become unstable. The upper object may be placed with a slight front-back offset relative to the supporting object, causing it to tilt or slip after release. This failure mainly results from inaccurate front-back displacement estimation from the front-view camera, where small offsets along this direction are difficult to capture reliably. Although HarmoWAM can preserve the overall stacking intention and approach trajectory, stable stacking still requires highly accurate relative placement, where minor visual localization errors can lead to post-release instability.

Insertion Misalignment. In the flower insertion task, the handover and transport stages can succeed, but the final insertion stage may still fail as the stem brushes against the vase rim. This failure is mainly caused by slight variations in the grasping position, which change the effective exposed length of the stem and make the insertion pose harder to align. Since this task requires bimanual handover, accurate relative pose recovery, and tight-tolerance insertion, it remains challenging for all evaluated models. Even when HarmoWAM maintains the correct task progress and moves the flower toward the vase, small grasp-induced pose deviations can prevent smooth insertion into the vase opening.

Zipper Grasp Slip. During the zipper-pulling process, HarmoWAM may fail when the zipper slips out of the gripper. This failure is mainly caused by the material and mechanical properties of the gripper, which may not provide sufficient friction or stable contact to hold the small zipper pull during continuous pulling. Since zipper manipulation requires sustained force along a constrained direction, even a slight loss of contact can break the pulling process and prevent successful closure.

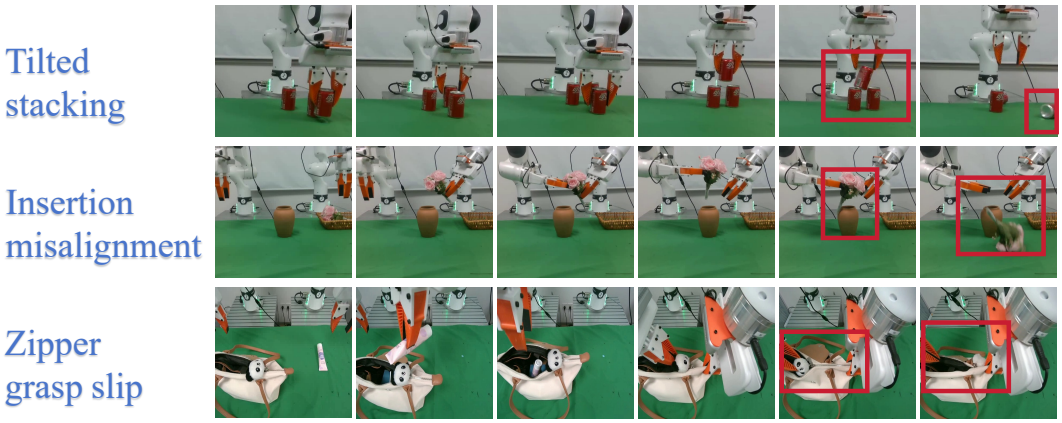


Figure 12: **Failure case visualization of HarmoWAM.** We visualize representative failures in real-world Franka experiments, with red boxes highlighting the key error regions.

H BROADER IMPACT

HarmoWAM aims to improve robotic manipulation by leveraging world-model-driven physical reasoning to coordinate predictive and reactive control. By enabling robust transit and precise interaction in unseen environments, our method reduces reliance on task-specific data collection and environment-specific retraining, supporting robot deployment beyond controlled laboratories. This capability can benefit applications such as smart manufacturing, logistics automation, assistive robotics, and domestic service, where robots must handle diverse backgrounds, changing object positions, and novel object semantics. However, deploying robotic policies in physical environments may introduce risks, including unexpected behaviors caused by inaccurate predictions, out-of-distribution observations, or ambiguous instructions. Therefore, real-world deployment should incorporate safety guardrails, uncertainty-aware control, human supervision, and hardware-level constraints. Overall, HarmoWAM contributes to more reliable robotic manipulation, helping robots move from laboratory demonstrations to complex, unstructured real-world environments.

I LIMITATIONS

The current implementation of HarmoWAM uses a fixed video prediction horizon, limiting its flexibility in accommodating tasks with different temporal requirements. Future work will explore adaptive prediction horizons based on task context and execution progress. Moreover, while pixel-level future generation provides intuitive visual guidance, it may introduce unnecessary generation overhead. Future work will further investigate latent-level predictive representations, enabling more efficient downstream action generation.