ImageWAM: Do World Action Models Really Need Video Generation, or Just Image Editing?

Yuyang Zhang [123*], **Wenyao Zhang** [123*†], **Zekun Qi** [4], **He Zhang** [3], **Haitao Lin** [3], **Jingbo Zhang** [3], **Yao Mu** [1], **Xiaokang Yang** [1], **Wenjun Zeng** [2], **Xin Jin** [25✉]

[1]Shanghai Jiao Tong University, [2]Eastern Institute of Technology, [3]Tencent Robotics X, [4]Tsinghua University, [5]Zhongguancun Academy
*Equal contribution, †Project Lead, ✉Corresponding author

World Action Models (WAMs) commonly rely on video generation to bridge visual world modeling and robot control. However, video-based WAMs face three coupled limitations: dense multi-frame future tokens make inference costly, full video prediction spends capacity on action-irrelevant temporal and appearance details, and long-horizon future imagination may introduce errors that mislead action prediction. These issues raise a simple question: *Does world action model really need video generation*? We propose **ImageWAM**, a simple WAM framework that repurposes pretrained image editing models for robot action prediction. In contrast to video generation, image editing provides a better-matched prior: it only needs to model a target-frame transformation, focuses on action-relevant current-to-target visual differences, and grounds task instructions to localized visual changes through edit pretraining. In practice, ImageWAM does not decode the target frame at inference time; instead, it conditions a flow-matching action expert on the KV caches produced by image-editing denoising, using them as a compact world-action context. ImageWAM outperforms standard VLA baselines and matching competitive WAMs without additional policy pretraining across different simulator and real-world experiments. It also reduces FLOPs to 1/6 and latency to 1/4 of video-based WAMs. Attention analysis further shows that editing caches focus on task-relevant change regions, supporting image editing as an effective alternative to video-based world-action modeling.

Date: June 19, 2026
Project Page: https://zhangwenyao1.github.io/ImageWAM/
Github: https://github.com/yuyangalin/ImageWAM

1 Introduction

Recent robot policy learning has increasingly explored video generation models as world-action backbones. This direction is appealing because video pretraining exposes models to rich visual dynamics, such as object motion, temporal continuity, physical interaction, and scene evolution [1–5]. It also supports a reason-before-act paradigm: a policy may first imagine how the scene will change, and then use this imagined future to guide action prediction [6–8]. Together with the scalability of generative pretraining on large and heterogeneous video data [9–12], video models provide an intuitive bridge between visual world modeling and robot control.

However, this bridge also reveals a mismatch as shown in Figure 1(a). Video generation models are trained to synthesize complete future videos. To do so, they must model appearance details, background changes, camera motion, temporal smoothness, and many other factors that may be only weakly related to the robot's next action [13–15]. Generating many spatio-temporal tokens across multiple frames makes inference costly for real-time robot control [2, 3]. Moreover, generating a physically consistent video is a hard proxy task [16–18]. This is especially true for fine-grained manipulation, where small contact events, slight object displacements, or subtle configuration changes can determine success, but are difficult to predict reliably over multiple frames. If the imagined video is wrong, the downstream action predictor may be misled. These issues raise a simple question: *Does the world action model really require video generation?*

We argue that image editing models offer a more direct visual generative prior for language-conditioned manipulation. Instead of predicting how an entire scene evolves over time, image editing models are trained to transform a source image according to a language instruction. This objective matches a key requirement of

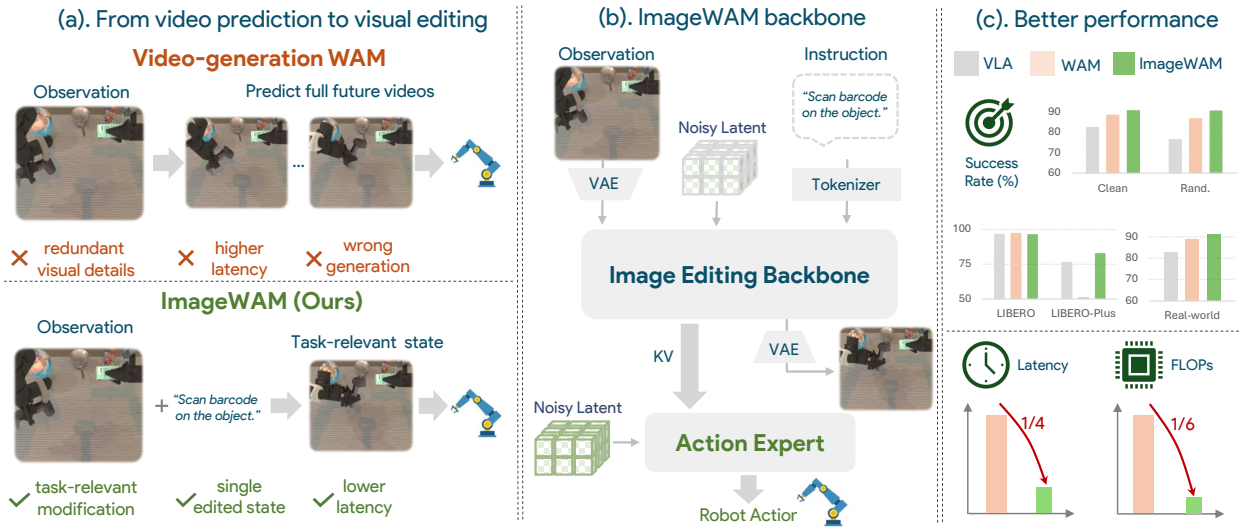

Figure 1 Previous video-generation WAMs instantiate world-action reasoning by predicting dense future video tokens, which can be computationally expensive and may allocate capacity to action-irrelevant visual details. ImageWAM replaces future video prediction with an image editing backbone that reasons over a source-grounded, instruction-guided visual transformation. The resulting edit-aware representation serves as a compact world-action intermediate for action prediction, achieving strong policy performance while reducing inference cost.

robot policies: the model should understand what task-relevant visual change should happen in the current scene under the given instruction. For many manipulation tasks, the essential signal is not a photorealistic future video, but an instruction-guided transformation from the current observation toward a desired visual state as illustrated in Figure 1(a).

This view gives image editing models three advantages as robot policy backbones. First, they provide strong instruction-to-change alignment. Their pretraining objective directly couples language with visual modifications, encouraging the model to focus on what should change, where it should change, and how the change is specified by the instruction. Second, editing provides an easier and more action-relevant proxy than full video prediction. Rather than modeling complete temporal trajectories, an editing model focuses on the visual difference between the current state and an instruction-consistent target state. This avoids spending capacity on irrelevant temporal details and reduces the risk of using inaccurate future videos for action generation. Third, editing offers a more compact inference path. A policy can use internal editing-aware representations that encode the intended visual transformation, without decoding dense multi-frame videos at inference time.

Motivated by this insight, we propose **ImageWAM**, a new framework that repurposes pretrained image editing models as backbones for robot action prediction, as shown in Figure 1(b). Given the current observation and task instruction, ImageWAM extracts editing-aware representations from an image editing backbone and feeds them into an action prediction head. Our goal is not to generate visually appealing edited images, nor to use editing models as goal-image generators. Instead, we use their intermediate instruction-conditioned features as transformation-aware representations for direct policy learning. This design preserves the benefits of generative visual pretraining while avoiding explicit future video synthesis, leading to a compact inference path for real-time control.

Empirically, we find that editing-aware representations are effective for language-conditioned robot policies. Under comparable action prediction architectures, ImageWAM improves over standard visual and vision-language backbones, showing that the gains are not merely due to stronger image recognition or language alignment. Our analyses further show that instruction conditioning and editing-oriented feature extraction are important for obtaining action-relevant representations. These results suggest that image editing models provide a promising backbone choice for robot policy learning, broadening visual generative pretraining beyond video-based world modeling.

Our contributions are three-fold:

- We introduce ImageWAM, a framework that repurposes pretrained image editing models as instruction-conditioned visual backbones for robot action prediction, offering an alternative to video-generation-based world action models.
- We formulate robot manipulation as instruction-guided visual transformation and identify three properties of image editing pretraining that are well aligned with policy learning: instruction-to-change alignment, easier goal/change proxy, and compact inference.
- We empirically validate the effectiveness of editing-aware representations against standard visual and vision-language backbones, and analyze the role of instruction conditioning and editing-oriented feature extraction in action prediction.

2 Related Works

2.1 Image Editing

Text-guided image editing modifies a source image according to a language instruction while preserving irrelevant content [19–28]. Recent diffusion-based and MLLM-enhanced editing models have progressed from simple object-level edits to more complex spatial, semantic, and knowledge-driven modifications [29–35]. While prior work mainly focuses on perceptual quality and instruction fidelity, we study image editing from a robotics perspective, using its source-conditioned and change-centric representations as compact world-action backbones for robot policy learning.

2.2 World Action Models

Unlike vision language action models [36–57], video generation models have recently been explored as predictive priors for robot policy learning. Early world action model [58–61] treats video generation as an explicit visual planning model: given the current observation and task context, the model predicts a complete future video or visual rollout, which is then translated into executable actions by an inverse dynamics model or action decoder [62–68]. More recent works broaden this paradigm by using video generative models as representation extractors for action generation [5, 69–78], value prediction [79] and interactive world modeling [80–83]. However, they are still largely built around video generation priors. Such designs often require predicting or processing dense spatio-temporal future tokens, leading to non-trivial inference cost and potentially modeling action-irrelevant and unrealistic visual details. ImageWAM uses instruction-guided editing caches as a compact world-action context, avoiding dense future-video token processing while preserving the advantage of WAMs.

3 Method

3.1 Problem Formulation

We consider robot manipulation conditioned on a current visual observation and a task instruction. At each time step t, the robot receives an image observation o_t and a task instruction l, and predicts an action chunk

$$\mathbf{a}_{t:t+H} = (a_t, a_{t+1}, \ldots, a_{t+H}), \tag{1}$$

where H denotes the action horizon. The policy learning objective is

$$\pi_\theta(\mathbf{a}_{t:t+H} \mid o_t, l). \tag{2}$$

World-action models introduce an intermediate visual reasoning step before action prediction. Video-generation-based WAMs typically instantiate this intermediate by predicting a future visual trajectory:

$$(o_t, l) \rightarrow \hat{o}_{t+1:t+H+1} \rightarrow \mathbf{a}_{t:t+H}. \tag{3}$$

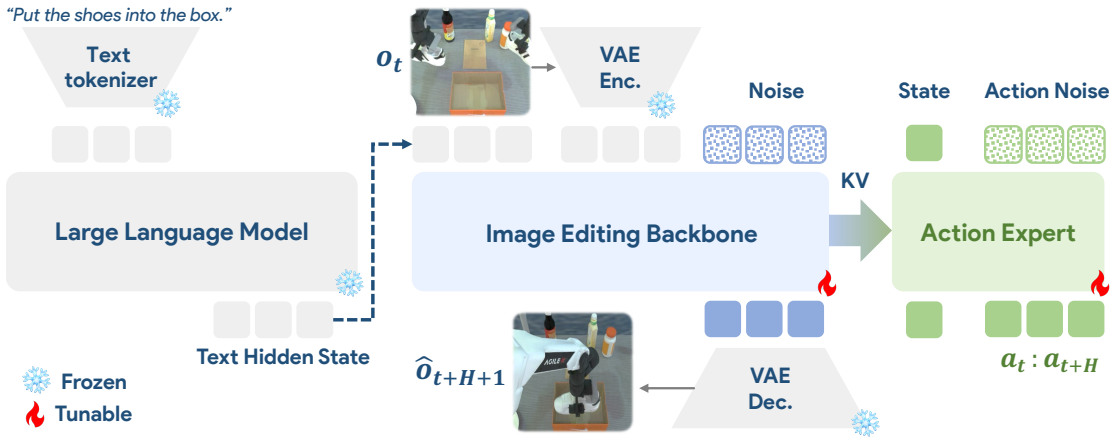

Figure 2 ImageWAM Pipeline. Given a language instruction and the current observation o_t, the image editing backbone synthesizes the future frame $\hat{o}_{t+H+1}$. The Action Expert integrates the intermediate KV features from this generation process via joint attention, predicting a sequence of future actions $\mathbf{a}_{t:t+H}$ conditioned on the current robot state and action noise.

This enables reason-before-act policy learning, but requires generating dense spatio-temporal visual tokens across multiple future frames. Instead of predicting the full future trajectory, Our ImageWAM predicts only the endpoint frame:

$$(o_t, l) \rightarrow \hat{o}_{\text{edit}} \equiv \hat{o}_{t+H+1} \rightarrow \mathbf{a}_{t:t+H}. \tag{4}$$

$\hat{o}_{\text{edit}}$ is a single source-conditioned frame that summarizes the task-specified visual transformation of the current observation. It serves as a compact world-action intermediate for action prediction.

3.2 ImageWAM Architecture

ImageWAM builds on a variant image editing model like OmniGen2 [84],Ovis-U1 [85] and Flux2 [86] by attaching an action expert to their image editing branch. OmniGen2 provides a source-conditioned image editing backbone that takes the current observation o_t and task instruction l as inputs. Instead of using the editing branch only to decode an edited image, ImageWAM reuses the intermediate transformer key-value caches produced during denoising as conditioning context for action generation.

During training, we randomly sample an editing denoising timestep τ and run the editing branch at this timestep. For each transformer layer ℓ, we collect the corresponding key-value cache:

$$\mathcal{C}^{\tau}_{\text{edit}} = \{(K^{\tau}_{\ell}, V^{\tau}_{\ell})\}_{\ell=1}^{L} = f^{\tau}_{\text{edit}}(o_t, l), \tag{5}$$

where L is the number of transformer layers. The cache $\mathcal{C}^{\tau}_{\text{edit}}$ is computed after the visual latent has interacted with the task instruction through the editing backbone. It therefore contains task-conditioned visual transformation information without requiring the final edited image to be decoded.

The action expert conditions on $\mathcal{C}^{\tau}_{\text{edit}}$ for action generation. This design transfers the image editing model's internal reasoning process to robot control: the editing branch reasons about how the source observation should change under the task instruction, while the action expert converts this editing context into executable robot actions. Unlike video-generation WAMs, ImageWAM does not require future video tokens to be generated or decoded.

In addition to the standard video-WAM variant that performs denoising over future video tokens, we also implement a Fast-WAM-style variant [13]. In this variant, future video tokens are used only during training for video co-training, but are removed at inference time. The action expert is conditioned on the KV caches produced from the current observation and task instruction, without instantiating or denoising future video tokens. This gives a video-WAM baseline with the same no-future-token inference interface as Fast-WAM.

We keep the VLM and multimodal understanding components of the editing model frozen, including the modules used to encode task instructions and visual context. Only the diffusion-based image generation branch and the action expert are updated during training. The frozen VLM provides stable language-vision conditioning, while the trainable diffusion branch learns to predict task-relevant future frames and to produce editing caches useful for action generation.

3.3 Action Prediction and Training

Image editing objective. The editing branch is trained to predict a task-relevant future endpoint frame. Let o_{t+H+1} denote the target future observation and let $z^*_{t+H+1} = E_{\text{vae}}(o_{t+H+1})$ be its latent representation. We sample image noise $\epsilon_z \sim \mathcal{N}(0, I)$ and an image flow time $r \in (0, 1)$, and construct the interpolated image latent

$$z_r = (1 - r)z^*_{t+H+1s} + r\epsilon_z. \tag{6}$$

The diffusion image branch predicts the corresponding velocity field:

$$\mathcal{L}_{\text{img}} = \mathbb{E}_{z^*,\epsilon_z,r}\left[\left\|u_\phi(z_r, r \mid o_t, l) - (\epsilon_z - z^*_{t+K})\right\|_2^2\right], \tag{7}$$

where u_ϕ denotes the velocity predictor of the diffusion image branch. This objective preserves the editing branch's ability to predict task-relevant future visual states and encourages the extracted editing caches to encode useful visual transformation information.

Action flow matching. The action expert generates an action chunk using a flow-matching objective. Let $\mathbf{a}^*_{t:t+H}$ denote the expert action chunk and let $\epsilon_a \sim \mathcal{N}(0, I)$ be Gaussian noise. We sample an action flow time $s \in (0, 1)$ and construct the interpolated action sample

$$\mathbf{a}_s = (1 - s)\mathbf{a}^*_{t:t+H} + s\epsilon_a. \tag{8}$$

Conditioned on the current observation, task instruction, and editing context cache $\mathcal{C}^\tau_{\text{edit}}$, the action expert predicts the velocity field:

$$\mathcal{L}_{\text{act}} = \mathbb{E}_{\mathbf{a}^*,\epsilon_a,s,\tau}\left[\left\|v_\theta(\mathbf{a}_s, s \mid o_t, l, \mathcal{C}^\tau_{\text{edit}}) - (\epsilon_a - \mathbf{a}^*_{t:t+H})\right\|_2^2\right]. \tag{9}$$

Here, s denotes the action flow-matching time, while τ denotes the image editing denoising timestep used to extract the editing cache. Sampling τ during training exposes the action expert to editing caches from different stages of the denoising process. We jointly optimize the diffusion image branch and the action expert with $\mathcal{L} = \mathcal{L}_{\text{act}} + \mathcal{L}_{\text{img}}$.

3.4 Efficient Inference

At inference time, ImageWAM avoids full future-video generation and also does not require decoding a complete edited image. Instead of running the full image editing denoising trajectory, we select a fixed editing denoising timestep $\tau^\star$ and perform only one editing-branch forward step to obtain

$$\mathcal{C}^{\tau^\star}_{\text{edit}} = f^{\tau^\star}_{\text{edit}}(o_t, l). \tag{10}$$

Action expert generates the action chunk by denoising action samples conditioned on this cache:

$$\hat{\mathbf{a}}_{t:t+H} \sim p_\theta(\mathbf{a}_{t:t+H} \mid o_t, l, \mathcal{C}^{\tau^\star}_{\text{edit}}). \tag{11}$$

This inference procedure is more compact than video-generation-based WAMs. A video WAM typically denoises and decodes dense spatio-temporal tokens across multiple future frames. In contrast, ImageWAM computes a single set of layer-wise editing caches and uses them directly as context for the action expert. Thus, ImageWAM preserves the reason-before-act principle of WAMs while avoiding the instantiation of dense future-video tokens.

For comparison, we also implement a Fast-WAM-style inference strategy for the video-WAM backbone. In this setting, future video tokens are removed at test time. The video backbone only processes the current observation and task instruction, and the action expert uses the resulting current-context KV caches for action generation. Therefore, this variant keeps a compact action-conditioning interface but avoids future-video token denoising during deployment.

4 Experiments

4.1 Experiment Setup

Unlike many VLA and WAM baselines that rely on additional embodied policy pretraining (**P.T.**), ImageWAM does not use extra embodied data and is trained only on the downstream benchmark demonstrations. We evaluate ImageWAM on LIBERO [87], LIBERO-Plus [88] and RoboTwin 2.0 [89], as well as on several real-world manipulation tasks as shown in Figure 3 with Flux.2 4B.

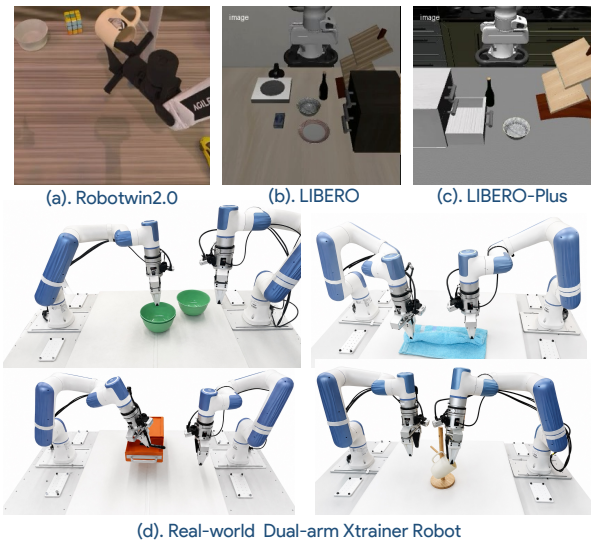

Figure 3 Experiments setup on Robotwin2.0, LIBERO, LIBERO-Plus and real-world robot.

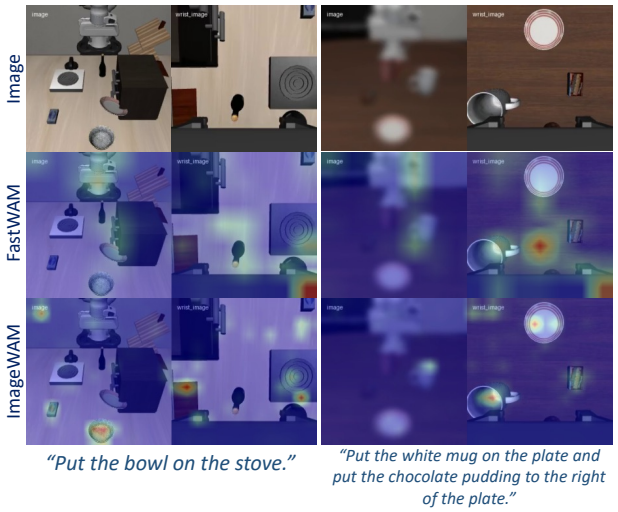

Figure 4 Attention visualization.

Table 1 Results on RoboTwin2.0.

Method	P.T.	Clean	Rand.	**Avg.**
π_0 [36]	✓	65.92	58.40	62.16
$\pi_{0.5}$ [37]	✓	82.74	76.76	79.75
ABot-M0 [90]	✗	81.20	80.40	80.80
Motus [12]	✓	88.66	87.02	87.80
LingBot-VA [3]	✓	92.90	91.50	92.20
FastWAM [13]	✗	91.88	91.78	91.83
ImageWAM	✗	**93.20**	**93.56**	**93.38**

Table 2 Results on LIBERO.

Method	P.T.	Spatial	Object	Goal	Long	**Avg.**
OpenVLA [91]	✓	84.7	88.4	79.2	53.7	76.5
GR00T N1 [38]	✓	84.7	88.4	79.2	53.7	76.5
π_0 [36]	✓	96.8	98.8	95.8	85.2	94.1
$\pi_{0.5}$ [37]	✓	**98.8**	98.2	98.0	92.4	96.9
LingBot-VA [3]	✓	98.5	99.6	97.2	98.5	**98.5**
Motus [12]	✓	96.8	99.8	96.6	97.6	97.7
Fast-WAM [13]	✗	98.2	**100.0**	97.0	95.2	97.6
ImageWAM	✗	97.2	99.2	**98.8**	98.4	98.4

LIBERO & LIBERO-Plus. We evaluate our model on LIBERO [92] and LIBERO-Plus [88]. For LIBERO, we follow the standard benchmarking protocol and train on the four standard suites: Spatial, Object, Goal and LIBERO-Long. Each suite contains 500 expert demonstrations spanning 10 tasks.

LIBERO-Plus provides a more challenging evaluation setting built upon the LIBERO tasks, with increased visual and layout variations. Following prior work, we use the same original LIBERO training demonstrations and do not incorporate the augmented LIBERO-Plus training data. We evaluate the trained policies under the LIBERO-Plus protocol and report the average success rate.

RoboTwin 2.0. We further evaluate on RoboTwin 2.0 [89], a large-scale simulated benchmark for bimanual robot manipulation. The benchmark covers more than 50 tasks and requires policies to coordinate two robot arms under diverse object layouts and scene conditions. Following the multi-task setting used in prior work [3, 13], we train a single policy on demonstrations from all tasks, including 2,500 trajectories collected in clean scenes and 25,000 trajectories collected with heavy scene randomization. All models are trained for 30k steps. We evaluate each method under both clean and randomized test settings, and report the average success rate over 100 trials per task.

Table 3 Comparison on the LIBERO-Plus benchmark. We report the average success rate across each perturbation dimension, where each perturbation includes the four task suites.

Method	LIBERO-Plus								
	P.T.	Camera	Robot	Language	Light	Background	Noise	Layout	Avg
UniVLA [93]	✓	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
OpenVLA-OFT [94]	✓	56.4	31.9	79.5	88.7	**93.3**	75.8	74.2	69.6
π_0 [36]	✓	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
π_0-Fast [95]	✓	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6
WorldVLA [96]	✓	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
FastWAM [13]	✗	16.4	44.5	68.9	78.2	53.7	37.7	60.7	51.5
ImageWAM(Omnigen2)	✗	80.0	49.2	70.9	82.6	69.4	77.1	71.8	71.8
ImageWAM(Ovis-U1)	✗	63.3	**58.4**	75.4	86.3	66.7	75.2	74.6	71.2
ImageWAM(FLUX.2 4B)	✗	**80.8**	50.3	**91.4**	**98.1**	85.5	**93.8**	**80.5**	**83.1**

Real-world Experiments. We also evaluated our model in a real-world dual-arm robot setup. We used the Dobot XTrainer dual-arm robotic platform to collect a dataset consisting of four tasks: **Stack Three Bowls(T1), Fold Towel(T2), Open Drawer & Store Marker(T3), and Hang Cup On Rack(T4)**. These tasks involve long-horizon manipulation, visual occlusion, fine-grained manipulation, and deformable-object manipulation, allowing us to assess the real-world performance of the model. Each task contains 100 trajectories. The model was trained on the combined dataset across all tasks, and all models were trained for 30k steps. We report the overall success rate over 100 trials conducted under multiple different initial configurations on this platform.

4.2 Main Results

Results on RoboTwin 2.0. Table 1 reports the results on RoboTwin 2.0 under both clean and randomized evaluation settings. In the clean setting, ImageWAM achieves an average success rate of 93.20%. In the randomized setting, ImageWAM achieves an average success rate of 93.56%. Compared with VLA baselines, ImageWAM shows a clear improvement, indicating that the editing-based world-action context provides useful visual transformation information for multi-task control. Compared with video-generation-based WAMs, ImageWAM reaches comparable performance while avoiding dense future-video token prediction, leading to a more efficient world-action reasoning pathway.

Results on LIBERO & LIBERO-Plus. Table 2 summarizes the results on LIBERO. On the standard LIBERO benchmark, ImageWAM achieves strong performance across Spatial, Object, Goal, and Long suites, showing that the editing-based backbone is effective for diverse manipulation skills. ImageWAM obtains an average success rate of 98.4%, remaining competitive with video-generation-based WAMs and pretrained VLA without any data pretraining.

Under the LIBERO-Plus setting, ImageWAM maintains an average success rate of 83.1%. This suggests that the source-conditioned editing context helps the policy focus on task-relevant visual changes rather than overfitting to fixed visual configurations. Together, the results on LIBERO and LIBERO-Plus indicate that image-editing-based world-action reasoning generalizes well across both standard and distribution-shifted simulation benchmarks.

Results on Real-world. As shown in Table 4, ImageWAM achieves an average success rate of 84.5%, outperforming π_0 (55.8%), $\pi_{0.5}$ (72.3%), and FastWAM (79.0%). Notably, ImageWAM performs best on all four real-world tasks, covering long-horizon manipulation, deformable-object manipulation, visual occlusion, and fine-grained control. Compared with FastWAM, ImageWAM improves success rates by 6 points on T1 (Stack Three Bowls), 9 points on T2 (Fold Towel), 1 point on T3 (Open Drawer & Store Marker), and 6 points on T4 (Hang Cup On Rack). The largest gain appears on T2, suggesting that the editing-based context is particularly useful when the task requires reasoning about task-relevant visual changes in deformable-object manipulation. On T3, both WAM-style methods substantially outperform π_0, indicating that world-action reasoning helps mitigate the impact of visual occlusion during manipulation. Overall, these results show that replacing dense video-token reasoning with image-editing caches yields a practical and efficient WAM backbone.

Table 4 Real-robot eval. Success rates (%).

Method	T1	T2	T3	T4	Avg
π_0 [36]	57	58	54	54	55.8
$\pi_{0.5}$ [37]	83	77	74	55	72.3
FastWAM [13]	88	75	77	76	79.0
ImageWAM(Ours)	**94**	**84**	**78**	**82**	**84.5**

Table 5 Efficiency. Lower is better.

Method	Lat.	TFLOPs	Interm.
FastWAM-IDM	1081 ms	63.65	Video
FastWAM (1 Step)	302 ms	13.21	Cache
ImageWAM(Ours)	**263 ms**	**9.72**	Cache

4.3 Analysis

Attention Visualization. Figure 4 visualizes the attention maps from the ImageWAM and FastWAM. ImageWAM concentrates attention on task-relevant change regions, including manipulated objects, target receptacles, and contact areas, while suppressing irrelevant background regions. This indicates that the editing caches encode source-grounded and change-centric visual information, providing useful context for the action expert.

Latency and FLOPs. Table 5 compares inference latency and FLOPs on A6000 GPU. Video-generation WAMs process dense spatio-temporal tokens across multiple future frames, whereas ImageWAM obtains a single set of image-editing caches from one editing-branch forward step. As a result, ImageWAM reduces latency from 1081 ms to 263 ms and FLOPs from 63.65 to 9.7, while maintaining competitive task success. This demonstrates that editing caches offer a more efficient world-action intermediate than future-video token rollout.

Qualitative analysis of future-video artifacts. Figure 5 illustrates a failure case of video-generation-based WAMs. The imagined future frames contain visible artifacts around task-relevant objects, including distorted geometry and inconsistent spatial layout. Such artifacts may mislead the action expert, since the predicted action is conditioned on the generated future representation. In contrast, ImageWAM does not instantiate dense future-video tokens or decode future frames at inference time. It directly uses image-editing caches as compact action-conditioning context, avoiding the accumulation of visual artifacts in future-video imagination.

4.4 Ablation Study

Q1: Can we use different editing models? We evaluate whether ImageWAM depends on a specific editing backbone by replacing OmniGen2 [84] with Ovis-U1 [85] and FLUX.2 4B [86], while keeping the action expert and training data unchanged. As shown in Table 7, all variants outperform FastWAM and most VLA baselines on LIBERO-Plus without policy pretraining. OmniGen2 and Ovis-U1 achieve similar average success rates of 71.8% and 71.2%, respectively, while FLUX.2 4B further improves the average to 83.1% and performs best on most perturbation dimensions. These results show that ImageWAM is not tied to a particular edit model, and that stronger editing backbones can directly improve policy robustness.

Q2: Why do we not use unified understanding-and-generation models?

Unified multimodal models that combine understanding and generation are promising, but the two capabilities impose different architectural demands. Understanding benefits from high-level semantic abstraction, whereas generation requires fine-grained spatial and structural details, especially in deeper layers [98]. Jointly optimizing both objectives in a single fully shared model may therefore introduce interference, where improving generation can hurt understanding, and vice versa. Instead, ImageWAM decouples these roles: we keep the VLM-based understanding components frozen and adapt only the diffusion generation branch and the action expert for robot control. As shown in Table 6, this design outperforms UniVLA and BagelVLA under similar non-keyframe future prediction setting, which are built upon unified understanding-and-generation models, while requiring no additional policy pretraining.

Table 6 Comparison with unified understanding-and-generation models. K.F. denotes keyframe prediction instead of plain future prediction which we adopt.

Method	P.T.	LIBERO	RoboTwin2.0 Clean Only	RoboTwin2.0 Clean2Hard
UniVLA [97]	✓	95.5	–	–
BagelVLA (w/ K.F.) [6]	✓	–	75.3	**20.9**
BagelVLA (w/o K.F.) [6]	✓	–	56.7	15.9
ImageWAM (Ours)	✗	**98.4**	**84.4**	18.3

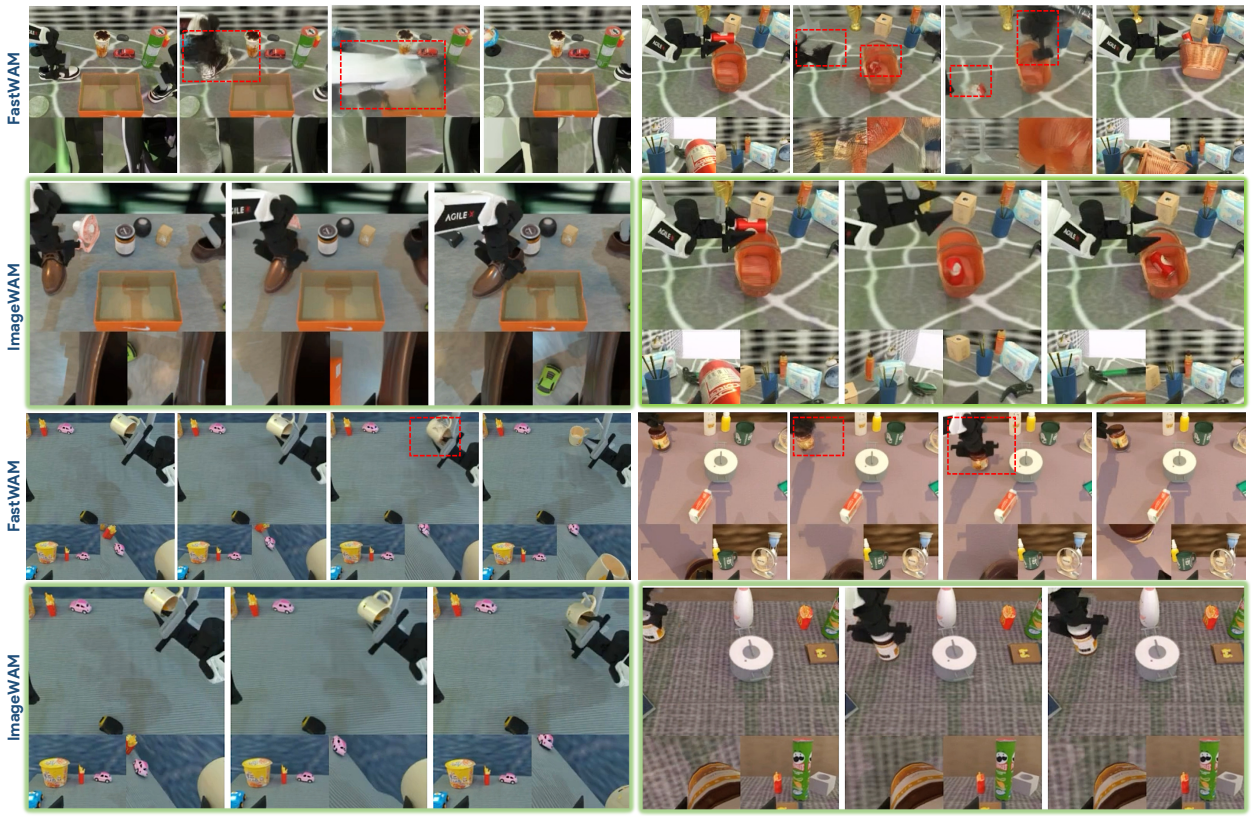

Figure 5 Future-video artifacts can mislead action prediction. The video-WAM baseline generates distorted future observations around task-relevant objects, leading to an unreliable action-conditioning context and task failure. ImageWAM avoids dense imagination and instead conditions the action expert on compact image-editing caches.

Q3: What is the effect of the size of the editing backbone? We evaluate whether increasing the capacity of the editing backbone improves the robustness of the policy in LIBERO-Plus. Replacing FLUX.2 4B with a larger FLUX.2 backbone increases the average success rate from 83.1% to 85.21%. The improvement mainly comes from Robot, Language, Background, and Layout perturbations, suggesting that larger editing models provide stronger instruction-conditioned visual context for action prediction. However, the gains are not uniform across all dimensions: Camera, Light, and Noise do not improve monotonically. This indicates that backbone scaling generally improves robustness, but the benefit depends on how the editing cache aligns with different perturbation types.

Table 7 Effect of using a larger editing backbone on LIBERO-Plus. We report the average success rate across each perturbation dimension, where each dimension includes the four LIBERO task suites.

Method	LIBERO-Plus								
	P.T.	Camera	Robot	Language	Light	Background	Noise	Layout	Avg
ImageWAM(FLUX.2 4B)	✗	80.8	50.3	91.4	98.1	85.5	93.8	80.5	83.1
ImageWAM(FLUX.2 9B)	✗	79.8	58.7	95.2	96.1	91.2	93.3	83.1	85.2

5 Conclusion

In this paper, we explore employing an image editing rather than a video generation model as the WAM backbone because image editing is an inherently ideal general task that naturally demands both visual understanding and generation. By simply predicting a single future frame, our model provides strong intermediate representations for the action model and enables end-to-end policy learning. Our model achieves a 93.56% success rate on RoboTwin (Random), substantially outperforming all VLA baselines and reaching performance comparable to state-of-the-art WAM models. We argue that the language-vision interaction priors in editing models drive our model's effectiveness and lay the groundwork for broader use of image models.

References

[1] 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. *arXiv preprint*, 2024.

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

[3] 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.

[4] Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. Dit4dit: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2025.

[5] 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.

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

[7] Jianke Zhang, Yuanfei Luo, Yucheng Hu, Xiaoyu Chen, Yanjiang Guo, Ziyang Liu, Hongbin Xu, Tian Lan, and Jianyu Chen. Uam: A dual-stream perspective on forgetting in vla training. *arXiv preprint arXiv:2605.15735*, 2026.

[8] Liaoyuan Fan, Zetian Xu, Chen Cao, Wenyao Zhang, Mingqi Yuan, and Jiayu Chen. Aim: Intent-aware unified world action modeling with spatial value maps. *arXiv preprint arXiv:2604.11135*, 2026.

[9] 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. *arXiv preprint*, 2025.

[10] Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, et al. Lda-1b: Scaling latent dynamics action model via universal embodied data ingestion. *arXiv preprint arXiv:2602.12215*, 2026.

[11] Wenyao Zhang, Bozhou Zhang, Zekun Qi, Wenjun Zeng, Xin Jin, and Li Zhang. Disentangled robot learning via separate forward and inverse dynamics pretraining. *arXiv preprint arXiv:2604.16391*, 2026.

[12] 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.

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

[14] 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*, 2026.

[15] Hanyang Yu, Haitao Lin, Jingbo Zhang, Wenyao Zhang, Chenghao Gu, Heng Li, and Ping Tan. Maskwam: Unifying mask prompting and prediction for world-action models. *arXiv preprint arXiv:2606.13515*, 2026.

[16] Baorui Peng, Wenyao Zhang, Liang Xu, Zekun Qi, Jiazhao Zhang, Hongsi Liu, Wenjun Zeng, and Xin Jin. Reworld: Multi-dimensional reward modeling for embodied world models. *arXiv preprint arXiv:2601.12428*, 2026.

[17] Xiuyu Yang, Bohan Li, Shaocong Xu, Nan Wang, Chongjie Ye, Zhaoxi Chen, Minghan Qin, Yikang Ding, Zheng Zhu, Xin Jin, et al. Orv: 4d occupancy-centric robot video generation. *arXiv preprint arXiv:2506.03079*, 2025.

[18] Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Tesseract: Learning 4d embodied world models. 2025. URL https://arxiv.org/abs/2504.20995.

[19] Yunnan Wang, Ziqiang Li, Wenyao Zhang, Zequn Zhang, Baao Xie, Xihui Liu, Wenjun Zeng, and Xin Jin. Scene graph disentanglement and composition for generalizable complex image generation. *Advances in Neural Information Processing Systems*, 37:98478–98504, 2024.

[20] Google DeepMind. Nano banana pro. https://deepmind.google/technologies/gemini/, 2025. Built on Gem-

ini 3 Pro. Image generation and editing model.

[21] OpenAI. GPT-Image-1.5. https://openai.com/index/new-chatgpt-images-is-here/, 2026. Accessed: 2026-03-19.

[22] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*, 2025.

[23] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[24] Zhipu AI. Glm-image. https://huggingface.co/zai-org/GLM-Image, 2026.

[25] NextStep Team, Chunrui Han, Guopeng Li, Jingwei Wu, Quan Sun, Yan Cai, Yuang Peng, Zheng Ge, Deyu Zhou, Haomiao Tang, et al. Nextstep-1: Toward autoregressive image generation with continuous tokens at scale. *arXiv preprint arXiv:2508.10711*, 2025.

[26] Meituan LongCat Team, Bin Xiao, Chao Wang, Chengjiang Li, Chi Zhang, Chong Peng, Hang Yu, Hao Yang, Haonan Yan, Haoze Sun, et al. Longcat-next: Lexicalizing modalities as discrete tokens. *arXiv preprint arXiv:2603.27538*, 2026.

[27] Dian Zheng, Manyuan Zhang, Hongyu Li, Hongbo Liu, Kai Zou, Kaituo Feng, and Hongsheng Li. Uni-edit: Intelligent editing is a general task for unified model tuning. *arXiv preprint arXiv:2605.21487*, 2026.

[28] Z-Image Team. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

[29] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In *Advances in Neural Information Processing Systems*, 2023.

[30] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. In *International Conference on Learning Representations*, 2024.

[31] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8871–8879, 2024.

[32] Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26125–26135, 2025.

[33] Valentin Gabeur, Shangbang Long, Songyou Peng, Paul Voigtlaender, Shuyang Sun, Yanan Bao, Karen Truong, Zhicheng Wang, Wenlei Zhou, Jonathan T Barron, et al. Image generators are generalist vision learners. *arXiv preprint arXiv:2604.20329*, 2026.

[34] Haoxiao Wang, Antao Xiang, Haiyang Sun, Peilin Sun, Changhao Pan, Yifu Chen, Minjie Hong, Weijie Wang, Shuang Chen, Yue Chen, et al. Diffusion model as a generalist segmentation learner. *arXiv preprint arXiv:2604.24575*, 2026.

[35] Gabriel Jeanson, David-Alexandre Duclos, William Larrivée-Hardy, Noé Cochet, Matěj Boxan, Anthony Deschênes, François Pomerleau, and Philippe Giguere. Leveraging image generators to address training data scarcity: The gen4regen dataset for forest regeneration mapping. *arXiv preprint arXiv:2605.05627*, 2026.

[36] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint*, 2024.

[37] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. pi0.5: a vision-language-action model with open-world generalization. *arXiv preprint*, 2025.

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

[39] Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: A vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint*, 2025.

[40] Wenxuan Song, Ziyang Zhou, Han Zhao, Jiayi Chen, Pengxiang Ding, Haodong Yan, Yuxin Huang, Feilong Tang, Donglin Wang, and Haoang Li. Reconvla: Reconstructive vision-language-action model as effective robot perceiver. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18549–18557, 2026.

[41] HY Team, Xumin Yu, Zuyan Liu, Ziyi Wang, He Zhang, Yongming Rao, Fangfu Liu, Yani Zhang, Ruowen Zhao, Oran Wang, et al. Hy-embodied-0.5: Embodied foundation models for real-world agents. *arXiv preprint arXiv:2604.07430*, 2026.

[42] Haitao Lin, Hanyang Yu, Jingshun Huang, He Zhang, Yonggen Ling, Ping Tan, Xiangyang Xue, and Yanwei Fu. Universal pose pretraining for generalizable vision-language-action policies. *arXiv preprint arXiv:2602.19710*, 2026.

[43] Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Zhuoguang Chen, Tao Jiang, and Hang Zhao. Depthvla: Enhancing vision-language-action models with depth-aware spatial reasoning. *arXiv preprint arXiv:2510.13375*, 2025.

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

[45] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. *ICLR*, 2024.

[46] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. *arXiv preprint*, 2025.

[47] Wanpeng Zhang, Ye Wang, Hao Luo, Haoqi Yuan, Yicheng Feng, Sipeng Zheng, Qin Jin, and Zongqing Lu. Dig-flow: Discrepancy-guided flow matching for robust vla models. *arXiv preprint arXiv:2512.01715*, 2025.

[48] Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-h0: Vision-language-action pretraining from large-scale human videos. In *International Conference on Machine Learning*. PMLR, 2026.

[49] Jiayi Chen, Wenxuan Song, Pengxiang Ding, Ziyang Zhou, Han Zhao, Feilong Tang, Donglin Wang, and Haoang Li. Unified diffusion vla: Vision-language-action model via joint discrete denoising diffusion process. *arXiv preprint arXiv:2511.01718*, 2025.

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

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

[52] Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, et al. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 40, pages 18638–18646, 2026.

[53] Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026.

[54] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.

[55] Qi Lv, Weijie Kong, Hao Li, Jia Zeng, Zherui Qiu, Delin Qu, Haoming Song, Qizhi Chen, Xiang Deng, and Jiangmiao Pang. F1: A vision-language-action model bridging understanding and generation to actions. *ArXiv*, abs/2509.06951, 2025. URL https://api.semanticscholar.org/CorpusID:281204333.

[56] Qiuyue Wang, Mingsheng Li, Jian Guan, Jinhui Ye, Sicheng Xie, Yitao Liu, Junhao Chen, Zhixuan Liang, Jie Zhang, Xintong Hu, et al. Qwen-vla: Unifying vision-language-action modeling across tasks, environments, and robot embodiments. *arXiv preprint arXiv:2605.30280*, 2026.

[57] Kechun Xu, Zhenjie Zhu, Anzhe Chen, Shuqi Zhao, Qing Huang, Yifei Yang, Haojian Lu, Rong Xiong, Masayoshi Tomizuka, and Yue Wang. Seeing to act, prompting to specify: A bayesian factorization of vision language action policy. *arXiv preprint arXiv:2512.11218*, 2025.

[58] 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. *NeurIPS*, 2024.

[59] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint*, 2023.

[60] Yao Feng, Hengkai Tan, Xinyi Mao, Guodong Liu, Shuhe Huang, Chendong Xiang, Hang Su, and Jun Zhu. Generalist bimanual manipulation via foundation video diffusion models. *arXiv preprint*, 2025.

[61] Youpeng Wen, Junfan Lin, Yi Zhu, Jianhua Han, Hang Xu, Shen Zhao, and Xiaodan Liang. Vidman: Exploiting implicit dynamics from video diffusion model for effective robot manipulation. *NeurIPS*, 2024.

[62] Yuejiang Liu, Fan Feng, Lingjing Kong, Weifeng Lu, Jinzhou Tang, Kun Zhang, Kevin P. Murphy, Chelsea Finn, and Yilun Du. World action verifier: Self-improving world models via forward-inverse asymmetry. 2026. URL https://api.semanticscholar.org/CorpusID:287074218.

[63] Boyuan Chen, Tianyuan Zhang, Haoran Geng, Kiwhan Song, Caiyi Zhang, Peihao Li, William T. Freeman, Jitendra Malik, Pieter Abbeel, Russ Tedrake, Vincent Sitzmann, and Yilun Du. Large video planner enables generalizable robot control. *ArXiv*, abs/2512.15840, 2025. URL https://api.semanticscholar.org/CorpusID:283933826.

[64] 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*, 2025.

[65] 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*, abs/2601.18323, 2026. URL https://api.semanticscholar.org/CorpusID:285051517.

[66] Zhongrui Zhang, Cheng-Chuan Yang, Qin Lu, Yanjiang Guo, Jianke Zhang, Yucheng Hu, and Jianyu Chen. Veo-act: How far can frontier video models advance generalizable robot manipulation? 2026. URL https://api.semanticscholar.org/CorpusID:287202336.

[67] Zirui Ge, Pengxiang Ding, Baohua Yin, Qishen Wang, Zhiyong Xie, Yemin Wang, Jinbo Wang, Hengtao Li, Runze Suo, Wenxuan Song, et al. Vampo: Policy optimization for improving visual dynamics in video action models. *arXiv preprint arXiv:2603.19370*, 2026.

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

[69] Yaxuan Li, Yichen Zhu, Junjie Wen, Chaomin Shen, and Yi Xu. Worldeval: World model as real-world robot policies evaluator. *arXiv preprint arXiv:2505.19017*, 2025.

[70] Mutian Xu, Tianbao Zhang, Tianqi Liu, Zhaoxi Chen, Xiaoguang Han, and Ziwei Liu. Kinema4d: Kinematic 4d world modeling for spatiotemporal embodied simulation. *arXiv preprint arXiv:2603.16669*, 2026.

[71] Zhennan Jiang, Shangqing Zhou, Yutong Jiang, Zefang Huang, Mingjie Wei, Yuhui Chen, Tianxing Zhou, Zhen Guo, Hao Lin, Quanlu Zhang, et al. Wovr: World models as reliable simulators for post-training vla policies with rl. *arXiv preprint arXiv:2602.13977*, 2026.

[72] Ruicheng Zhang, Guangyu Chen, Zunnan Xu, Zihao Liu, Zhizhou Zhong, Mingyang Zhang, Jun Zhou, and Xiu Li. Robostereo: Dual-tower 4d embodied world models for unified policy optimization. *arXiv preprint arXiv:2603.12639*, 2026.

[73] Boyu Chen, Yi Chen, Lu Qiu, Jerry Bai, Yuying Ge, and Yixiao Ge. Unit: Toward a unified physical language for human-to-humanoid policy learning and world modeling. *arXiv preprint arXiv:2604.19734*, 2026.

[74] Jai Bardhan, Patrik Drozdik, Josef Sivic, and Vladimir Petrik. Persistent robot world models: Stabilizing multi-step rollouts via reinforcement learning. *arXiv preprint arXiv:2603.25685*, 2026.

[75] Bingchuan Wei, Bingqi Huang, Jingheng Ma, Sen Cui, et al. Fate: Closed-loop feasibility-aware task generation with active repair for physically grounded robotic curricula. *arXiv preprint arXiv:2603.01505*, 2026.

[76] Xiaolei Lang, Yang Wang, Yukun Zhou, Chaojun Ni, Kerui Li, Jiagang Zhu, Tianze Liu, Jiajun Lv, Xingxing Zuo, Yun Ye, et al. Vag: Dual-stream video-action generation for embodied data synthesis. *arXiv preprint arXiv:2604.09330*, 2026.

[77] Yixuan Wang, Rhythm Syed, Fangyu Wu, Mengchao Zhang, Aykut Onol, Jose Barreiros, Hooshang Nayyeri, Tony Dear, Huan Zhang, and Yunzhu Li. Interactive world simulator for robot policy training and evaluation. *arXiv preprint arXiv:2603.08546*, 2026.

[78] Yuejiang Liu, Fan Feng, Lingjing Kong, Weifeng Lu, Jinzhou Tang, Kun Zhang, Kevin Murphy, Chelsea Finn, and Yilun Du. World action verifier: Self-improving world models via forward-inverse asymmetry. *arXiv preprint arXiv:2604.01985*, 2026.

[79] Runze Li, Hongyin Zhang, Junxi Jin, Qixin Zeng, Zifeng Zhuang, Yiqi Tang, Shangke Lyu, and Donglin Wang. World-value-action model: Implicit planning for vision-language-action systems. *arXiv preprint arXiv:2604.14732*, 2026.

[80] Yue Liao, Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Hu Yue, 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. *ArXiv*, abs/2508.05635, 2025. URL https://api.semanticscholar.org/CorpusID:280545868.

[81] Yaxuan Li, Zhongyi Zhou, Ye Chen, Yaokai Xue, and Yichen Zhu. dworldeval: Scalable robotic policy evaluation via discrete diffusion world model. 2026. URL https://api.semanticscholar.org/CorpusID:287773839.

[82] Yixuan Wang, Rhythm Syed, Fangyu Wu, Mengchao Zhang, Aykut Onol, Jose Barreiros, Hooshang Nayyeri, Tony Dear, Huan Zhang, and Yunzhu Li. Interactive world simulator for robot policy training and evaluation. 2026. URL https://api.semanticscholar.org/CorpusID:286377674.

[83] Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

[84] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025.

[85] Guo-Hua Wang, Shanshan Zhao, Xinjie Zhang, Liangfu Cao, Pengxin Zhan, Lunhao Duan, Shiyin Lu, Minghao Fu, Jianshan Zhao, Yang Li, and Qing-Guo Chen. Ovis-u1 technical report. *arXiv preprint arXiv:2506.23044*, 2025.

[86] Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

[87] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint*, 2023.

[88] Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

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

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

[91] 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*, 2024.

[92] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: benchmarking knowledge transfer for lifelong robot learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *NeurIPS*, 2023.

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

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

[95] 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*, 2025.

[96] 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*, 2025.

[97] Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. *arXiv preprint arXiv:2506.19850*, 2025.

[98] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

Appendix

5.1 Architecture of ImageWAM

Across the three model variants, namely OmniGen2, FLUX.2[klein], and Ovis-U1, we adopt the MoT structure as our multimodal joint modeling architecture.

5.1.1 OmniGen2-based ImageWAM

For the OmniGen2-based ImageWAM variant, we load the LLM component from the corresponding original pretrained Qwen2.5-VL-3B as the LLM backbone, which provides the downstream model with a strong foundation for vision-language alignment. The last-layer hidden states of the Qwen2.5-VL LLM are fed into the OmniGen2 DiT, together with the latent tokens of the reference image and the future noisy frames, for self-attention. In MoT, we extend the original self-attention mechanism into joint self-attention over four types of tokens: language context tokens, visual condition tokens, visual prediction tokens, and action tokens. The visual prediction transformer and the action transformer independently generate their attention QKV representations, which are then concatenated into a complete QKV sequence. The attention mask is configured such that action tokens attend to the other tokens in a one-way manner, while noisy tokens attend only to context tokens, thereby keeping the information in the context tokens clean.

To prevent the visual model from being affected by noisy gradients from the action model during the early stage of training, we adopt an action-head weight-copy initialization strategy similar to [3, 13]. Specifically, our Action DiT uses the same architecture as the image editing model. We copy and interpolate the weights of the image editing model to match the size of the Action DiT, and add additional projection layers to support action inputs and outputs. To enable cross-modal attention while maintaining a moderately sized Action DiT, we use a relatively small DiT hidden dimension 1024 while keeping the same attention hidden dimension 2520. The final size of our Action DiT is approximately 760M parameters.

5.1.2 FLUX.2-based ImageWAM

For the FLUX.2-based architecture, the LLM module is the original pretrained Qwen3-4B/8B used by FLUX.2. We similarly extend FLUX.2 into a joint self-attention structure, while modifying the action-head initialization strategy according to the double-stream and single-stream design of FLUX.2. In this setting, the lower layers of the action head are initialized by copying the weights from the image stream in the double-stream stage of FLUX, while the higher layers are initialized from the single-stream blocks of FLUX. The final sizes of the Action DiT in this variant are 642M parameters for the 4B version and 952M parameters for the 9B version.

5.1.3 Ovis-U1-based ImageWAM

For the Ovis-U1-based architecture, we use the Qwen3-1.7B model trained and vision-language fine-tuned by Ovis-U1, and adopt its approximately 1.2B-parameter diffusion visual decoder as our visual editing backbone. In this model, the language context tokens also include vision-language tokens processed by the LLM. Since Ovis-U1 adopts an MMDiT structure similar to FLUX, we use the same Action DiT initialization strategy as in the FLUX.2-based ImageWAM variant. Because this model is relatively small, we do not reduce the DiT hidden dimension. The final size of the Action DiT is 1.1B parameters.

5.2 Training Details

All models are trained on 8 NVIDIA H20 GPUs. Unless otherwise specified, we use DeepSpeed ZeRO-1 for distributed training. For the FLUX.2 9B variant, we use DeepSpeed ZeRO-2 due to its larger model size. All models are trained with bf16 precision and optimized using AdamW. The common training hyperparameters are summarized in Table 8.

On LIBERO, we horizontally concatenate the two camera views and resize the resulting image to 224×448. The model predicts the future observation 16 frames ahead, together with an action chunk of length 16. We train on the merged dataset of the four LIBERO suites for 10 epochs.

Table 8 Common training hyperparameters.

Parameter	Value
GPUs	8 NVIDIA H20
Distributed strategy	DeepSpeed ZeRO-1*
Precision	bf16
Optimizer	AdamW
Optimizer betas	$(0.9, 0.95)$
Learning rate	1×10^{-4}
Weight decay	1×10^{-2}
LR scheduler	Warmup cosine
Warmup steps	$0.05T_{\text{total}}$
Minimum LR	$0.01 \times \text{lr}$
Gradient clipping	1.0

*For FLUX.2 9B, we use ZeRO-2 for VRAM compatibility.

On RoboTwin, we first resize the two wrist-view images to a smaller resolution and horizontally concatenate them. The concatenated wrist views are then vertically concatenated with the main-view image, and the final input is resized to 288×256. The model also predicts the future observation 16 frames ahead and an action chunk of length 16. We train the models for 5 epochs.

On Real-World Dataset, we follow the same preprocess in RoboTwin, predicting 16 action steps and training on all four task for 10 epoch.

Table 9 Dataset-specific training configurations.

Parameter	LIBERO	RoboTwin
Input views	2 views	3 views
View layout	Horizontal	Wrist-horizontal + vertical
Input resolution	224×448	288×256
Future horizon	16 frames	16 frames
Action chunk length	16	16
Training epochs	10	5

Table 10 Training cost and batch size.

Benchmark	Model	Time	Batch/GPU
LIBERO	OmniGen2	18 hours	12
LIBERO	Ovis-U1	18 hours	16
LIBERO	FLUX.2 4B	18 hours	10
LIBERO	FLUX.2 9B	1.6 days	12
RoboTwin	OmniGen2	5 days	48†
RoboTwin	FLUX.2 4B	5 days	48†
Real-World Robot	OmniGen2	18 hours	16

† Effective per-GPU batch size with gradient accumulation over three steps.

6 Efficiency Optimization

To further optimize inference latency, we also evaluate on our model the prefix-only attention training and image-denoising-free inference strategy, similar to that adopted in FastWAM. In addition, we explore model optimization with 'torch.compile' and static CUDA graphs. The inference latency results are reported in Table 11, where all models use three action denoising steps during inference. We observe that adding compilation brings nearly a 3× overall speedup, mainly due to the improved efficiency of the action head. This is because, under typical action chunk lengths, the number of action tokens is relatively small, making the

parallel efficiency of the Action DiT often suboptimal.

Table 11 Inference latency and relative speedup. Speedup is computed with respect to FastWAM with one video denoising step.

Variant	Latency (ms)	Speedup
FastWAM (1× Vid. Denoise)	302	1.00×
ImageWAM (1× Vid. Denoise)	263	1.15×
FastWAM (Prefix Only)	194	1.56×
+ Compiled	80	3.78×
ImageWAM (Prefix Only)	198	1.53×
+ Action Loop Compile	85	3.55×
+ Image Prefill Compile	77	3.92×
+ Action Static Graph	69	4.38×

7 Real-World Experiments Detail

7.1 Task settings and evaluation in Real-world Tasks

Task Settings. To evaluate the capability and generalizability of ImageWAM, we design four representative and challenging real-world manipulation tasks, including: (1) **Stack Three Bowls (T1)**, stacking three green nested bowls; (2) **Fold Towel (T2)**, folding a fabric towel; (3) **Open Drawer & Store Marker (T3)**, which involves opening a drawer, placing a marker inside, and closing the drawer; and (4) **Hang Cup On Rack (T4)**, hanging a mug onto a designated peg on a wooden stand. We collect an average of 100 demonstrations per task. Each model is evaluated over 50 trials per task. The execution success rate is reported as the primary performance metric.

8 RoboTwin Evaluation Results

Here we present the per-task results on RoboTwin evaluation in Table 12.

Table 12 Per-task success rates on RoboTwin under clean and randomized evaluation settings.

Task	**ImageWAM Flux.2 4B (Ours)**		**ImageWAM OmniGen2 (Ours)** (50 trials)		Fast-WAM-IDM		Fast-WAM w.o. co-train		LingBot-VA		$\pi_{0.5}$		Motus	
	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.
Adjust Bottle	**100**	99	**100**	**100**	94	99	98	**100**	90	94	**100**	99	89	93
Beat Block Hammer	98	**99**	**100**	98	98	98	80	92	96	98	96	93	95	88
Blocks Ranking RGB	96	**99**	**100**	96	**100**	**99**	88	86	99	98	92	85	99	97
Blocks Ranking Size	**96**	**100**	86	92	79	90	56	62	94	96	49	26	75	63
Click Alarmclock	98	**100**	**100**	**100**	98	**100**	**100**	98	99	**100**	98	89	**100**	**100**
Click Bell	**100**	**100**	**100**	**100**	**100**	96	**100**	**100**	**100**	**100**	99	66	**100**	**100**
Dump Bin Bigbin	**96**	90	92	88	93	**98**	92	94	89	96	92	97	95	91
Grab Roller	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Handover Block	96	**95**	94	84	97	94	58	46	**99**	78	66	57	86	73
Handover Mic	**100**	**100**	**100**	**100**	98	99	**100**	**100**	94	96	98	97	78	63
Hanging Mug	**74**	**84**	50	56	66	62	28	40	40	28	18	17	38	38
Lift Pot	**100**	**100**	**100**	**100**	**100**	**100**	92	90	**100**	99	96	85	96	99
Move Can Pot	96	98	96	92	**97**	**100**	80	68	94	97	51	55	34	74
Move Pillbottle Pad	98	**100**	98	98	98	**100**	88	96	**99**	99	84	61	93	96
Move Playingcard Away	**100**	99	**100**	**100**	99	**100**	94	96	**100**	99	96	84	**100**	96
Move Stapler Pad	67	60	74	82	89	**85**	64	78	**91**	79	56	42	83	**85**
Open Laptop	98	98	96	**100**	92	92	**100**	98	92	94	90	96	95	91
Open Microwave	97	**94**	**98**	82	54	53	46	52	82	86	34	77	95	91
Pick Diverse Bottles	84	88	84	**92**	87	89	58	62	89	82	81	71	**90**	91
Pick Dual Bottles	96	98	**100**	**100**	**100**	98	80	74	**100**	99	93	63	96	90
Place A2B Left	95	93	94	**100**	**97**	96	84	92	**97**	93	87	82	88	79
Place A2B Right	96	94	96	**98**	94	**98**	88	84	**97**	95	87	84	91	87
Place Bread Basket	96	92	90	94	91	**97**	74	76	**97**	95	77	64	91	94
Place Bread Skillet	90	89	92	90	90	**95**	**98**	84	95	90	85	66	86	83
Place Burger Fries	95	**100**	**100**	**100**	97	99	94	96	97	95	94	87	98	98
Place Can Basket	74	72	**82**	76	37	28	72	72	81	**84**	62	62	81	76
Place Cans Plasticbox	99	97	**100**	94	98	96	98	96	**100**	**99**	94	84	98	94
Place Container Plate	98	98	98	98	**100**	96	94	98	99	97	99	95	98	**99**
Place Dual Shoes	81	81	90	84	85	87	80	74	**94**	**89**	75	75	93	87
Place Empty Cup	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99	99	98
Place Fan	95	94	94	88	97	**95**	80	88	**99**	93	87	85	91	87
Place Mouse Pad	84	93	90	84	**97**	93	64	76	93	**96**	60	39	66	68
Place Object Basket	86	83	**92**	**90**	87	82	82	**90**	91	88	80	76	81	87
Place Object Scale	97	96	92	98	**99**	**99**	86	80	96	95	86	80	88	85
Place Object Stand	98	98	**100**	92	96	**100**	82	92	99	96	91	85	98	97
Place Phone Stand	**100**	**100**	98	98	99	99	90	92	97	97	81	81	87	86
Place Shoe	97	95	94	96	95	**98**	92	90	98	**98**	92	93	**99**	97
Press Stapler	**97**	**100**	90	94	50	57	80	80	85	82	87	83	93	98
Put Bottles Dustbin	**97**	91	92	**96**	**97**	92	78	88	87	91	84	79	81	79
Put Object Cabinet	91	89	90	**96**	**93**	90	88	84	85	87	80	79	88	71
Rotate QRcode	87	**92**	82	90	91	86	82	78	**96**	91	89	87	89	73
Scan Object	94	90	94	86	93	90	78	86	**96**	**91**	72	65	67	66
Shake Bottle	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	97	99	97	**100**	97
Shake Bottle Horizontally	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99	99	99	**100**	98
Stack Blocks Three	96	97	**100**	**100**	99	95	90	90	99	98	91	76	91	95
Stack Blocks Two	99	**100**	**100**	**100**	**100**	**100**	**100**	98	**100**	98	97	**100**	**100**	98
Stack Bowls Three	78	83	84	86	85	83	66	82	**86**	83	77	71	79	**87**
Stack Bowls Two	94	97	92	**98**	94	96	90	**98**	94	**98**	95	96	**98**	**98**
Stamp Seal	79	84	76	84	**99**	94	60	78	96	**97**	79	55	93	92
Turn Switch	77	**79**	54	70	59	74	66	66	44	45	62	54	**84**	78
Average	**93.20**	**93.56**	92.48	92.80	91.16	91.34	82.76	84.80	92.90	91.50	82.74	76.76	88.66	87.02