FLOWWAM: OPTICAL FLOW AS A UNIFIED ACTION REPRESENTATION FOR WORLD ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs) leverage pretrained video generators for both world modeling and action prediction. However, directly leveraging such video generators for control raises a new challenge: how to represent actions in a suitable form that aligns with pretrained video generators while carrying enough motion cues for accurate control. Existing numerical actions fail to satisfy the former, and prior visual action representations overlook the temporal motion structure across frames. We address this issue with **FlowWAM**, a dual-stream diffusion framework that adopts optical flow as a unified, video-native action representation. Rather than attaching actions to the generator through a separate action space, FlowWAM natively internalizes them as a parallel video stream. Since flow videos share the same format as RGB videos and encode rich per-pixel displacement, FlowWAM jointly models both within a shared pretrained video generator, naturally implementing two modes of WAMs. In policy mode, FlowWAM generates flow for action prediction, while in world-model mode, it uses target flow sequences to guide future video generation. Moreover, since learning flow requires no action labels, FlowWAM can leverage large-scale action-unlabeled videos for pretraining. In policy mode, FlowWAM outperforms both VLA and WAM baselines on RoboTwin (93.7% average success), with action-unlabeled video pretraining notably boosting generalization to unseen randomized scenes, and surpasses dynamic-manipulation specialists on DOMINO (41.3% success). In world-model mode, its dense motion conditioning achieves the highest overall score on WorldArena, bringing an 18.4% relative improvement in trajectory accuracy. More results are available at `https://anonymous-flowwam.github.io`.

1 INTRODUCTION

World Action Models (WAMs) (Ye et al., 2026; Zhu et al., 2025a; Yang et al., 2025a; Bi et al., 2025; Kim et al., 2026; Pai et al., 2025; Li et al., 2026a; Yuan et al., 2026) have recently emerged as a promising direction alongside Vision-Language-Action (VLA) policies (Black et al., 2024; Kim et al., 2025; Li et al., 2025) for manipulation. While VLA policies often benefit from semantic priors in image-language pretraining, WAMs leverage pretrained video generators that already encode motion and temporal dynamics. These motion-centric priors are particularly well-suited for manipulation, where success depends not only on grounding instructions to objects but also on capturing the continuous, frame-by-frame motion of both the robot and the environment. To use these video priors for control, an ideal action representation should fit the visual input format of pretrained video generators while preserving the cross-frame motion cues needed for control.

Existing WAMs have explored several forms of action representation, yielding substantial advancements in the field. Numerical action tokens (Ye et al., 2026; Li et al., 2026a; Pai et al., 2025; Yuan et al., 2026) are precise, but they lie outside the visual space of the video generator and must be connected to it through separate action modules. Latent actions (Bi et al., 2025; Ye et al., 2025) are learned from frame transitions, yet they are abstract and may not preserve the dense, spatially grounded motion cues needed for control. Recent image-space actions, such as action ray maps (Jiang et al., 2025), masks (Chen et al., 2026a), or multi-view action images (Li et al., 2026b) reduce this gap by rendering control signals into visual form. However, these visual conditions are largely static spatial cues that indicate where actions occur, rather than how each visible part of the embodiment moves across frames. As a result, they remain frame-level conditioning signals injected into the

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.

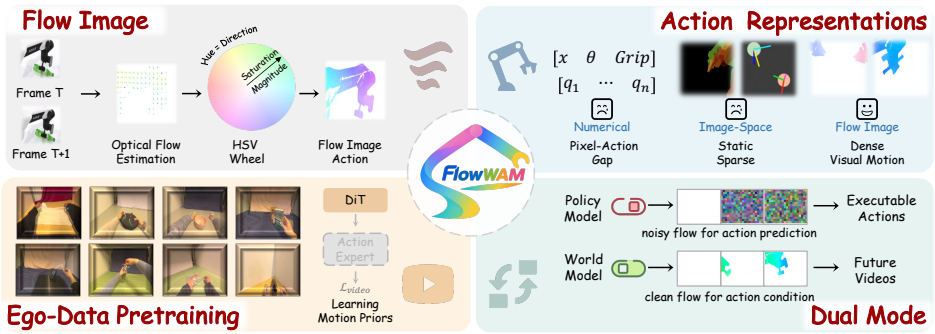

Figure 1: **FlowWAM uses optical flow as a unified action representation for world action models.** By representing actions with dense flow videos, FlowWAM bridges executable robot actions and pixel-space video priors, enabling action-unlabeled video pretraining, policy inference, and action-conditioned world modeling within the same framework.

video generator, rather than a temporally dense action representation that evolves jointly with the predicted future videos. These limitations leave an open question: **how can actions be represented in a video-native form that preserves dense cross-frame motion cues while remaining decodable into executable control?**

Our key observation is that **optical flow serves as a naturally suitable action representation for WAMs**. By recording per-pixel displacement and mapping it to the RGB space via HSV encoding, flow videos become dense, video-like motion signals that align seamlessly with the visual priors of pretrained video generators. Furthermore, because flow can be extracted without action labels, it provides a unified representation for both action-unlabeled pretraining and downstream control. While prior works treat flow as an auxiliary signal (Fang et al., 2025; Zhong et al., 2025) or a modular intermediate (Xu et al., 2025; Ranasinghe et al., 2026), we instead re-envision optical flow as a primary, video-native action modality that is deeply integrated into the WAM's generative latent space. Concretely, existing WAMs typically connect actions to the video generator through separate action spaces, relying on numerical tokens that lack visual grounding or latent codes that compress motion details. In contrast, FlowWAM models optical flow directly as a parallel, full-resolution video stream. This design avoids extra action encoders, preserves dense pixel-level motion details, and directly leverages the visual priors of pretrained video backbones.

Building on these insights, we propose **FlowWAM**, a dual-stream diffusion framework that unifies policy learning and world modeling within a single pretrained video backbone (Fig. 1). Because optical flow shares the standard video format with RGB frames, both modalities seamlessly share the pretrained VAE and transformer blocks with only lightweight adapters. This architecture establishes optical flow as a **unified action representation**, casting both robot control and world modeling into a consistent video generation paradigm. By natively embedding actions as a dense flow stream, FlowWAM achieves three key properties: (1) **Decodability for control:** predicted flow fields provide dense motion trajectories that are reliably translated into robot actions by an action expert (Sections 4.1, 4.2 and 4.4). (2) **Scalability via unlabeled pretraining:** flow extracted from egocentric videos transfers generalizable motion priors to downstream policy learning without requiring action annotations (Sections 4.1 and 4.5). (3) **Reliability for world modeling:** dense flow conditioning provides pixel-level motion guidance, steering future video generation with substantially higher trajectory fidelity than numerical actions (Section 4.3).

Our contributions are summarized as follows:

- We establish optical flow as a unified, video-native action representation for world action models, which aligns naturally with visual generation priors and unlocks scalable pretraining on action-unlabeled videos.
- We develop FlowWAM, a dual-stream diffusion framework that jointly generates RGB and flow videos using shared transformer blocks with lightweight adapters, coupled with an action expert that decodes predicted flow into executable robot control.

- We demonstrate FlowWAM across manipulation policy and world modeling benchmarks. FlowWAM sets state-of-the-art results on RoboTwin (93.7% average success) and the dynamic DOMINO benchmark (41.3% success), with pretraining substantially improving generalization to unseen scenes. On WorldArena, FlowWAM achieves a leading EWMScore of 63.71 with marked gains in trajectory accuracy.

2 Related Work

World Action Models. World Action Models (WAMs) adapt video generators for manipulation by coupling action prediction with world modeling. One line jointly generates videos and actions, as DreamZero (Ye et al., 2026) and Cosmos Policy (Kim et al., 2026) fine-tune video generators with action tokens and UWM (Zhu et al., 2025a) and CoVAR (Yang et al., 2025a) couple video and action diffusion heads through shared latents. Another line first generates future videos and then recovers actions with an inverse dynamics model (Hu et al., 2025; Pai et al., 2025; Feng et al., 2025b) or a causal prediction module (Li et al., 2026a). In both lines, actions remain a heterogeneous stream connected to video latents through action-specific tokens, heads, or decoders. FlowWAM instead represents action-relevant motion itself as a flow video, modeled as a visual stream inside the same pretrained video generator.

Action Representations in WAMs. Existing WAMs differ in how the action stream relates to the video generator. Numerical action tokens (Ye et al., 2026; Kim et al., 2026; Zhu et al., 2025a) keep actions in a separate symbolic space, and latent action codes learned from frame transitions (Ye et al., 2025; Tharwat et al., 2025) or compressed motion (Bi et al., 2025) are independent of specific robot embodiments. Image-space signals such as ray maps (Jiang et al., 2025), embodiment masks (Chen et al., 2026a), or multi-view action images (Li et al., 2026b) instead place action cues in the visual domain of the video generator. FlowWAM advances this image-space line by modeling dense optical flow at full resolution directly within the video generator. Unlike Motus (Bi et al., 2025), which compresses flow into an external latent space with a separate autoencoder, FlowWAM processes flow through shared VAE and transformers, serving as both the prediction target for policy control and the conditioning signal for world modeling.

Flow for Manipulation. Optical flow and pixel motion have also been used for manipulation, as auxiliary supervision or motion targets for VLA-style policies (Fang et al., 2025; Zhong et al., 2025; Bi et al., 2025), as intermediate variables in flow-to-action, planning, and future-flow prediction pipelines (Xu et al., 2025; Chen et al., 2025c; Gao et al., 2025; Ranasinghe et al., 2026), as structured pixel motion linking high-level generation and low-level control in LangToMo (Ranasinghe et al., 2025) and DAWN (Nguyen et al., 2025), or as sparse point tracks in ATM (Wen et al., 2023). For dynamic scenes, PUMA (Fang et al., 2026) feeds HSV-encoded optical flow from past frames to the visual encoder of a VLA, and the concurrent DynamicWAM (Lou et al., 2026) encodes past flow with a frozen video VAE as an extra condition of a WAM, alongside a separate kinematic pathway into its action expert. These methods treat flow as an auxiliary input, a modular intermediate, or an external planning signal. FlowWAM instead integrates dense flow natively alongside RGB within a shared video generator, so the same representation supports policy prediction and world modeling.

3 Method

FlowWAM uses optical flow as the unified action representation that connects pretrained video priors with executable robot control. Given a reference image I_0 and a language instruction τ, it jointly models RGB and flow videos with a dual-stream diffusion transformer. The same video generator supports two use modes. When flow is generated, FlowWAM acts as a policy whose motion plan is decoded into actions. When flow is provided, FlowWAM acts as a world model that renders an RGB future consistent with the specified motion. An overview of the full framework is shown in Fig. 2. We describe the dual-stream generation in Section 3.1, the action expert in Section 3.2, and the training procedure in Section 3.3.

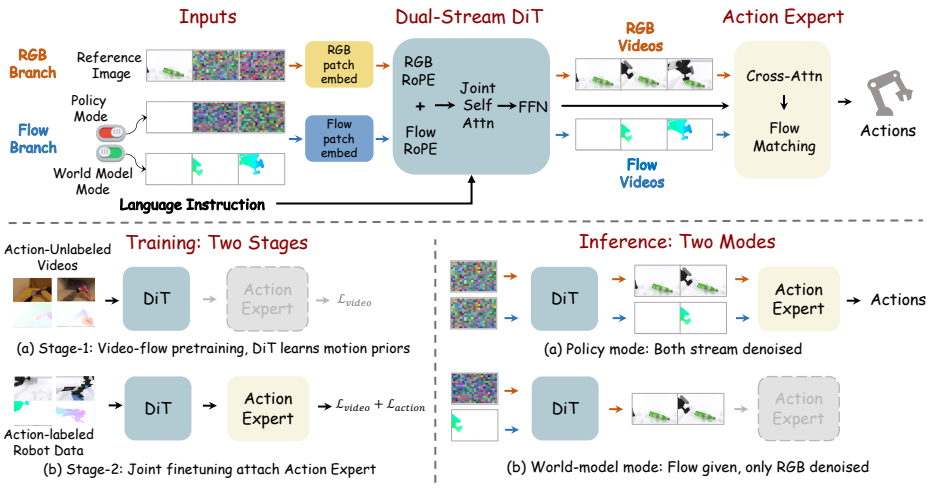

Figure 2: **Overview of FlowWAM.** FlowWAM encodes RGB frames and optical flow as two coupled video streams in a shared diffusion transformer.

3.1 Unified Video-Flow Generation

Flow RGB Encoding. We first convert action-relevant motion flow into the same image format as RGB video frames. Given consecutive frames, the optical flow field $\mathbf{f}_t \in \mathbb{R}^{H\times W\times 2}$ records per-pixel displacement (u, v), capturing both where motion occurs and how visible points move. We convert it into an RGB image with an HSV color-wheel encoding ϕ:

$$F_t = \phi(\mathbf{f}_t): \quad \mathrm{H} = \frac{\operatorname{atan2}(v,u)+\pi}{2\pi}, \quad \mathrm{S} = \frac{\|\mathbf{f}_t\|}{m}, \quad \mathrm{V} = 1, \tag{1}$$

where m is a normalization constant for the flow magnitude and H, S, and V denote the hue, saturation, and value channels, respectively. Since flow RGB F_t is format-identical to scene frames I_t, action-relevant motion can be directly processed by the same VAE encoder and video generator without a separate action tokenizer.

Dual-Stream Architecture. RGB and flow videos $\mathbf{V}$ and $\mathbf{F}$ are encoded by the same frozen VAE encoder $\mathcal{E}$ into latents $\mathbf{z} = \mathcal{E}(\mathbf{V})$ and $\mathbf{z}^f = \mathcal{E}(\mathbf{F})$. The two streams have their own patch embeddings and output heads but share all transformer blocks. Their tokens are concatenated for joint self-attention, so RGB and flow interact at every layer while the model retains the generative capacity of the pretrained video generator (Appendix A.1).

Two Operating Modes. Once RGB and flow are modeled as parallel streams, policy and world-model inference differ only in whether flow is unknown or observed. In both modes the first latent frame of each stream is fixed to the clean VAE encoding of the reference observation, and all subsequent latent frames are produced by iterative denoising. Motion history, when needed, is added the same way, by prepending past frames and their flow as further clean latents (Section 4.2).

In **policy mode**, the remaining frames of both streams are initialized from Gaussian noise and jointly denoised. The dual-stream DiT simultaneously synthesizes a future RGB rollout and its corresponding flow video, from which the action expert reads intermediate features to decode executable actions. In **world-model mode**, the flow latents $\mathbf{z}^f$ are instead set to the clean VAE encoding of a desired motion trajectory and held fixed throughout sampling, while only the RGB latents are initialized from noise. Thus, flow is the shared action representation in the framework. It is *generated* for policy inference and *provided* for controllable world modeling.

3.2 Action Prediction from Flow Latents

FlowWAM first predicts dense RGB and flow latents with the dual-stream DiT, and then translates this motion plan into the robot's low-level action space. We therefore use an action expert, a transformer

that cross-attends to the per-layer flow and RGB hidden states of the dual-stream video generator, conditions on the current proprioceptive state, and predicts an N-step action chunk $\hat{\mathbf{a}}_0 \in \mathbb{R}^{N \times d_a}$ under a flow-matching objective.

At training time the action expert can read hidden states computed from clean visual latents (the frozen VAE encoding). At inference it reads hidden states derived from the latents generated by the dual-stream model, which may contain residual denoising error. To align these distributions, we mix noise into the latents fed to the video generator on a fraction p=0.5 of training steps and pass the sampled noise level to the action expert as an auxiliary embedding,

$$\tilde{\mathbf{z}}^f = (1-\sigma)\,\mathbf{z}^f + \sigma\,\boldsymbol{\epsilon}^f, \quad \tilde{\mathbf{z}} = (1-\sigma)\,\mathbf{z} + \sigma\,\boldsymbol{\epsilon}^r, \quad \sigma \sim \mathcal{U}[0,1], \tag{2}$$

with independent $\boldsymbol{\epsilon}^f, \boldsymbol{\epsilon}^r \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The remaining $1-p$ training steps see pure clean latents.

3.3 Training

FlowWAM uses one training formulation for both action-labeled robot data and action-unlabeled video data. With action labels, the dual-stream video generator and action expert are optimized together. Without labels, the same RGB-flow video objective still trains the dual-stream video generator.

Video Generation Loss. The dual-stream DiT is trained with a flow matching objective (Lipman et al., 2023) over both RGB and flow latents. For joint RGB–flow generation, a shared timestep $t \sim \mathcal{U}[0,1]$ is sampled for both streams. The noisy latents are constructed as $\mathbf{z}_t = (1-t)\,\mathbf{z}_0 + t\,\boldsymbol{\epsilon}$ and $\mathbf{z}_t^f = (1-t)\,\mathbf{z}_0^f + t\,\boldsymbol{\epsilon}'$, and the model predicts the velocity fields $v_\theta(\mathbf{z}_t, t)$ and $v_\theta^f(\mathbf{z}_t^f, t)$. The video loss is a weighted combination of the two streams:

$$\mathcal{L}_{\text{video}} = (1-\lambda_f)\,\mathcal{L}_{\text{RGB}} + \lambda_f\,\mathcal{L}_{\text{flow}}, \tag{3}$$

where $\mathcal{L}_{\text{RGB}}$ and $\mathcal{L}_{\text{flow}}$ are the mean squared errors between predicted and target velocities for the RGB and flow streams, respectively. To match the image-to-video inference distribution, the first latent frame of each noisy stream is replaced with its clean conditioning latent, and the loss on this frame is correspondingly masked out. We additionally perturb the conditioning frame with a small Gaussian noise to mimic the residual error of the most recent observation in autoregressive rollouts. For world-model training and inference, we instead keep the flow stream at $t_f = 0$ as a clean condition, while applying the diffusion schedule only to the RGB stream.

Manipulation flow is spatially sparse, with motion concentrated around the embodiment, manipulated objects and contact-relevant regions. To avoid static-background dominance, we apply *motion-aware reweighting* to $\mathcal{L}_{\text{flow}}$. Per-location weights are computed from the channel-averaged deviation of the flow latent from the reference frame:

$$w_{\text{motion}} = 1 + \alpha \cdot \frac{\langle |\mathbf{z}^f - \mathbf{z}^f_{\text{ref}}| \rangle_c}{\max \langle |\mathbf{z}^f - \mathbf{z}^f_{\text{ref}}| \rangle_c}, \tag{4}$$

where $\langle \cdot \rangle_c$ averages over latent channels and α controls the boosting strength, directing learning toward motion-rich regions.

Action Prediction Loss and Total Objective. When action labels are available, the action expert is supervised to predict the ground-truth action chunk from the dual-stream DiT's per-layer hidden states under an independent flow-matching schedule, yielding a per-chunk loss $\mathcal{L}_{\text{action}}$. The labeled-data objective is

$$\mathcal{L} = \mathcal{L}_{\text{video}} + \lambda_a\,\mathcal{L}_{\text{action}}, \tag{5}$$

where λ_a balances visual motion modeling and action decoding.

Action-Free Pretraining on Unlabeled Videos. The same formulation also enables action-unlabeled video pretraining. Because $\mathcal{L}_{\text{video}}$ depends only on RGB frames and their optical flow, the dual-stream video generator can be trained on videos without action labels, such as egocentric human-manipulation videos. In practice, to narrow the embodiment gap between human videos and robot observations, we additionally convert EgoDex (Hoque et al., 2025) into robot videos by retargeting its tracked hands to 16 rendered robot arms, which provides robot-only flow in the same

Table 1: **RoboTwin 2.0 success rates (%), averaged over 50 tasks.** PT: pretraining.

	In-Domain (*Clean*+*Random*)			Out-of-Domain (*Clean*→*Random*)		
Method	***Clean***	***Random***	**Avg.**	***Clean***	***Random***	**Avg.**
VLA policies						
π_0 (Black et al., 2024)	65.92	58.40	62.16	46.42	16.34	31.38
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	82.74	76.76	79.75	70.66	46.00	58.33
X-VLA (Zheng et al., 2025)	72.88	72.84	72.86	67.96	20.95	44.46
ABot-M0 (Yang et al., 2026b)	86.06	85.08	85.57	57.40	30.36	43.88
StarVLA (StarVLA Community, 2026)	88.18	88.32	88.25	46.52	3.16	24.84
World action models						
Motus (Bi et al., 2025)	88.66	87.02	87.84	–	–	–
X-WAM (Guo et al., 2026a)	89.76	90.68	90.22	69.98	25.84	47.91
Fast-WAM (Yuan et al., 2026)	91.88	91.78	91.83	77.76	1.92	39.84
4D-WAM (Yang et al., 2026a)	91.92	90.76	91.34	81.50	41.80	61.65
LingBot-VA (Li et al., 2026a)	92.90	91.50	92.20	80.70	34.60	57.65
FlowWAM w/o PT	87.60	87.00	87.30	86.60	38.06	62.33
FlowWAM w/ PT	**93.74**	**93.68**	**93.71**	**89.24**	**53.04**	**71.14**

form as for robot data (Appendix A.4). The retargeted joint trajectories only drive this rendering and are never used as supervision, so pretraining remains action-free. We therefore adopt a two-stage recipe: (i) a *video-only* stage that updates only the dual-stream DiT under $\mathcal{L}_{\text{video}}$, and (ii) a *joint* stage on robot demonstrations that attaches the action expert and optimizes the full labeled-data objective. This decouples visual motion modeling from embodiment-specific action decoding, allowing motion priors learned from human video to transfer to robot control without changing the flow representation.

4 Experiments

We evaluate the three properties outlined in the introduction. Decodability for control is tested on RoboTwin 2.0 (Section 4.1), the dynamic DOMINO benchmark (Section 4.2), and real robots (Section 4.4). Scalability via action-unlabeled pretraining is tested through in-domain and out-of-domain evaluation on RoboTwin 2.0 (Section 4.1) and controlled ablations (Section 4.5), and reliability for world modeling is tested on WorldArena (Section 4.3). Section 4.5 also reports design ablations and inference cost.

4.1 Manipulation Policy

Setup. We evaluate manipulation policy on RoboTwin 2.0 (Chen et al., 2025a), which provides 50 bimanual tasks under a *Clean* setting with fixed layout and lighting and a *Random* setting with randomized object pose, distractors, lighting, and background. Models are trained on 50 demonstrations under *Clean* and 500 under *Random* together, and we report success rate over 100 rollouts per task. We refer to this as the *in-domain* protocol. We additionally evaluate an *out-of-domain* protocol, in which models are trained on the *Clean* demonstrations alone and evaluated on both settings, so the randomized scenes are never seen during training.

We compare against five direct-action VLA policies and five unified world action models that pair video prediction with action decoding (Table 1). To ensure a fair comparison, all WAM baselines are evaluated with their respective pretraining protocols on large-scale video or action-unlabeled data. FlowWAM's flow targets are rendered from the demonstrated joint trajectories (Appendix A.3), and it is reported with and without EgoDex pretraining (Section 3.3). Table 1 reports averages over all 50 tasks for both protocols, with per-task results for both in Appendix B.1.

Results. FlowWAM reaches 93.74% success on *Clean* and 93.68% on *Random* under the in-domain protocol in Table 1, the highest average success rate among all VLA and WAM baselines. The advantage is consistent with using flow as a video-native action stream. A dense image-space

Table 2: **Results on DOMINO** (clean, Level 1). SR: success rate (%). MS: manipulation score.

Method	SR ↑	MS ↑	Method	SR ↑	MS ↑
OpenVLA (Kim et al., 2024)	1.54	6.10	OpenVLA-OFT (Kim et al., 2025)	9.06	24.06
π_0-FAST (Pertsch et al., 2025)	3.54	20.87	$\pi_{0.5}$ (Physical Intelligence et al., 2025)	9.63	26.12
VLA-Adapter (Wang et al., 2026)	4.43	24.31	PUMA (Fang et al., 2026)	17.20	34.97
InternVLA-M1 (Chen et al., 2025b)	5.40	27.57	InternVLA-A1.5 (Ma et al., 2026)	29.30	42.50
Isaac-GR00T (Bjorck et al., 2025)	6.11	28.60	DynamicWAM (Lou et al., 2026)	38.20	53.16
π_0 (Black et al., 2024)	8.17	23.96	**FlowWAM (Ours)**	**41.31**	**55.26**

displacement field gives the action expert a more explicit motion plan than numerical actions, and is reliably decoded into joint actions rather than only improving video prediction.

Pretraining on action-unlabeled human egocentric videos further boosts FlowWAM under both settings in Table 1. Notably, even without video pretraining, FlowWAM achieves competitive performance against direct-action baselines, demonstrating that predicting dense optical flow provides an effective structured motion plan for control. We further isolate the specific contributions of the flow stream and pretraining data through controlled ablations in Section 4.5.

Generalization to unseen randomization. The right half of Table 1 presents the out-of-domain protocol, where models are trained exclusively on *Clean* demonstrations and evaluated on unseen randomized scenes. Most baselines suffer severe performance drops in this zero-shot setting. FlowWAM achieves 89.24% on *Clean* and 53.04% on *Random*, leading all compared methods in both columns and substantially outperforming existing world action models such as 4D-WAM (41.80%) and Fast-WAM (1.92%). Notably, action-unlabeled pretraining proves decisive on these unseen scenes, yielding a 14.98% absolute gain on Random and showing that pretraining on human videos provides essential visual robustness against novel environmental variations.

4.2 Dynamic Manipulation

Setup. We evaluate dynamic manipulation on DOMINO (Fang et al., 2026), a benchmark containing 35 tasks with moving targets. Because targets move continuously, policies must infer object dynamics from observation history rather than single frames. Following the standard protocol, we evaluate on the clean Level-1 split over 100 episodes per task and report the success rate (SR) and the continuous manipulation score (MS). We compare against nine general-purpose VLAs as well as dynamic-specialized methods, including PUMA (Fang et al., 2026) and DynamicWAM (Lou et al., 2026). FlowWAM ingests four past observation frames and their corresponding flows as clean latents (Section 3.1), with training details provided in Appendix C.1.

Results. As shown in Table 2, FlowWAM achieves 41.31% SR and 55.26 MS, outperforming all compared baselines. Standard VLAs struggle significantly in dynamic environments, with most falling below 10% success rate due to the lack of explicit temporal motion modeling. While dynamics-specific methods such as PUMA (17.20% SR) and DynamicWAM (38.20% SR) improve tracking through specialized history encoders or auxiliary kinematic pathways, FlowWAM surpasses them while maintaining a simpler architecture. FlowWAM obtains its motion history by placing past RGB frames and their robot-only flow directly into its two streams, without dedicated history encoders or multi-branch pipelines.

4.3 Action-Conditioned World Modeling

Setup. We evaluate action-conditioned world modeling on WorldArena (Shang et al., 2026), a unified benchmark that jointly scores video perception quality and action reliability over 121-frame rollouts at 24 fps. Each episode provides an initial frame, a language instruction, and the recorded joint-action trajectory, from which the model must synthesize the future video. FlowWAM builds its flow condition by rendering this trajectory with the robot model and camera geometry and estimating optical flow between the rendered robot-only frames, so no future ground-truth RGB frames are used. We report six representative metrics together with the overall EWMScore, with the full 16-metric breakdown in Appendix Table 15.

Table 3: **Main results on WorldArena. Best** and second-best per column are marked. *Cond.*: conditioning signal.

Method	Cond.	JEPA Sim. ↑	Subj. Cons. ↑	Bg. Cons. ↑	Traj. Acc. ↑	Depth Acc. ↑	Sem. Align. ↑	EWMScore ↑
General video foundation models								
CogVideoX (Yang et al., 2025b)	Text	94.84	80.97	88.38	34.79	91.09	89.70	58.79
Veo 3.1 (Google DeepMind, 2025)	Text	57.97	75.82	**91.67**	11.36	74.27	83.79	57.77
Wan 2.2 (Wan et al., 2025)	Text	72.02	72.91	79.37	16.93	80.57	88.98	51.71
Wan 2.6 (Wan Team, Tongyi Lab, Alibaba Group, 2025)	Text	72.57	73.15	84.29	12.18	75.43	88.09	59.80
Cosmos-Predict 2.5 (text) (Ali et al., 2025)	Text	46.69	66.19	73.98	11.60	70.71	86.34	53.06
ABot-PhysWorld (text) (Chen et al., 2026b)	Text	90.36	80.56	89.02	31.50	71.99	89.58	62.63
Action-conditioned robot world models								
Cosmos-Predict 2.5 (action) (Ali et al., 2025)	Action	90.65	67.24	72.72	27.49	90.35	89.86	54.29
IRASim (Zhu et al., 2025b)	Action	93.72	72.82	78.69	35.92	93.50	89.43	56.15
Ctrl-World (Guo et al., 2026b)	Action	92.77	**83.56**	90.30	48.20	93.25	88.68	59.98
Vidar (Feng et al., 2025a)	Img-Act	61.10	70.37	78.24	21.26	79.77	88.46	51.92
Genie Envisioner (Liao et al., 2025)	Img-Act	33.81	74.53	87.54	2.63	86.83	85.98	41.37
GigaWorld-1 (GigaWorld Team et al., 2026)	Img-Act	96.77	80.97	86.43	54.27	98.44	89.42	62.34
FlowWAM (Ours)	**Flow**	**97.14**	82.46	89.97	**64.26**	**98.97**	**89.93**	**63.71**

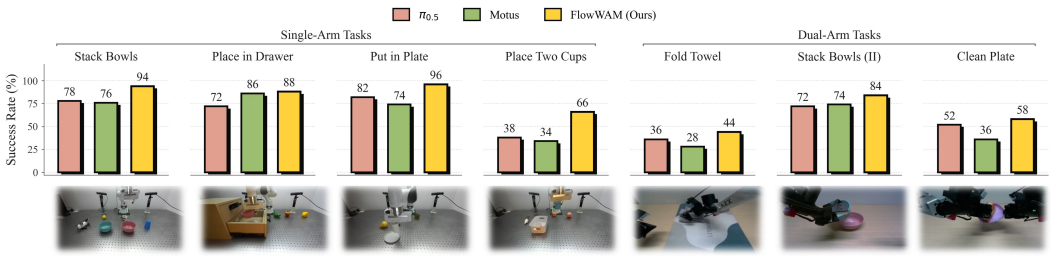

Figure 3: **Real-world success rates across single-arm and dual-arm manipulation tasks.**

Baselines in Table 3 cover general video and foundation world models, which test whether large-scale visual priors alone suffice for embodied prediction, and action-conditioned robot world models with numerical or image-space conditioning, which are designed for manipulation dynamics but do not adopt flow as the conditioning signal.

Results. FlowWAM achieves the best overall EWMScore of 63.71 in Table 3. The diagnostic axis for reliability is Trajectory Accuracy, which scores how faithfully a generated rollout follows the given action. Text- and numerical-action-conditioned baselines remain low in this column, image-space actions yield partial improvements, and FlowWAM leads the second-best method by nearly ten points. Wan 2.2 (Wan et al., 2025), the video model family on which FlowWAM is built, reaches only 16.93, so the gain does not come from the pretrained backbone alone. FlowWAM also remains close to the best baselines on Subject and Background Consistency and tops Depth Accuracy. Text, numerical, and image-action cues can preserve appearance but leave trajectory control under-specified, whereas flow provides a dense visual motion field, which explains why FlowWAM's largest margin appears on Trajectory Accuracy.

4.4 Real-World Manipulation

Setup. We further test whether the flow-based action representation transfers to real robots on seven tasks (Figure 3): four single-arm Franka tasks that stress spatial precision and object contact, and three dual-arm ARX tasks that additionally require bimanual coordination and longer-horizon motion (Appendix C.3). For each task, all methods are trained on the same 100 teleoperated demonstrations and evaluated over 50 trials with randomized object poses, with $\pi_{0.5}$ (Physical Intelligence et al., 2025) and Motus (Bi et al., 2025) as baselines.

Results. Figure 3 shows that FlowWAM reaches the highest success rate on every task, averaging 75.71% across the suite. The two baselines, $\pi_{0.5}$ and Motus, average 61.43% and 58.29%, respectively. The advantage holds on both platforms: FlowWAM averages 86.00% on the single-arm and 62.00% on the dual-arm tasks, compared with 67.50% and 53.33% for $\pi_{0.5}$ and 67.50% and 46.00% for

Table 4: **Ablations**. (a) Contribution of flow on RoboTwin (out-of-domain), where (ii)–(iv) share all other settings. IDM: inverse-dynamics decoder on the generated RGB video. (b) Policy-side design choices on RoboTwin (in-domain). (c) Action-conditioning variants on WorldArena, evaluated on a custom validation split because the official evaluation requires remote submission.

(a) Contribution of flow (RoboTwin).

	EgoDex pretraining	Decoder input	*Clean*	*Rand.*	Mean
(i)	None	RGB + flow	86.60	38.06	62.33
(ii)	RGB	RGB (IDM)	87.38	41.18	64.28
(iii)	RGB	RGB + flow	88.46	46.00	67.23
(iv)	RGB + flow	RGB + flow	**89.24**	**53.04**	**71.14**

(b) Policy mode (RoboTwin).

Variant	Succ. (%)
Numerical actions	71.89
Raw (u, v) flow	76.80
w/o flow-loss reweighting	88.30
w/o stochastic AE cond.	86.45
FlowWAM (full)	**93.71**

(c) World mode (WorldArena).

Conditioning	EWMScore
Text only	49.31
Numerical actions	54.18
Flow actions (u, v)	56.72
Image actions (masks)	57.84
FlowWAM (full)	**65.23**

Motus. Qualitative rollouts in Appendix D.2 (Figure 7) further visualize how the predicted flow plan aligns with robot trajectories on both platforms.

4.5 Analysis

Isolating the contribution of flow. To rigorously disentangle whether the performance gains originate from the flow representation, the pretraining data, or model capacity, Table 4(a) evaluates controlled variants under the out-of-domain protocol, sharing the identical 5B backbone, EgoDex video clips, and training schedule. First, under the same RGB-pretrained backbone, decoding actions through the generated optical flow stream (iii) outperforms an inverse-dynamics model reading generated RGB videos (ii) by 2.95% in mean success rate and 4.82% on unseen Random scenes, isolating the benefit of an explicit flow-based motion plan. Second, co-pretraining the flow stream alongside RGB on the same video data (iv) adds a further 3.91% in mean success rate and 7.04% on Random, confirming that the flow stream effectively acquires reusable motion priors from unlabeled videos. Finally, because the dual-stream adapters add only 1.2M parameters to the 5B backbone, these consistent gains are driven by the flow representation rather than added model capacity.

Ablation Study. Panels (b) and (c) of Table 4 ablate the design choices on both sides of the model. On the policy side (panel b), replacing flow with numerical action vectors collapses success to 71.89%, and substituting RGB flow with raw (u, v) tensors breaks the input format expected by the pretrained VAE and DiT (Eq. 1). Both results indicate that the flow channel itself, not generic visual conditioning, closes the action-video gap. Removing motion-aware reweighting (Eq. 4) lets the static background dominate $\mathcal{L}_{\text{flow}}$, while training the action expert without stochastic latent conditioning (Eq. 2) leaves it brittle to the residual denoising error of generated latents at inference.

On the world-model side (panel c), with the video generator fixed, text leaves motion under-specified, numerical actions lie outside the visual space, raw (u, v) tensors lack the image format of the pretrained generator, and image-space masks rendered from the same trajectory are largely static cues without cross-frame motion. Only RGB flow combines per-pixel motion with a video-native format, raising EWMScore from 49.31 to 65.23. Flow is thus required at both ends of the model, as a decodable target for actions and as a spatially aligned condition for generation.

Inference cost. With action-matching distillation and lossless optimizations such as KV caching and CUDA graphs, our default distilled version of FlowWAM on RoboTwin generates each action chunk in 352 ms on one NVIDIA H100 GPU (Appendix A.7).

5 Conclusion

In this paper, we introduced FlowWAM, which establishes optical flow as a unified action representation for WAMs. By encoding robot motion as dense flow videos, we bridge the modality gap between executable control and pretrained video priors, keeping the action inside the video generator rather than in a separate action space. This unified representation allows the same model to generate flow for policy decoding, condition on flow for world modeling, and learn from action-unlabeled egocentric videos. Experiments show that this representation improves both control and world modeling, and controlled ablations that hold the backbone and pretraining data fixed attribute these gains to the flow stream itself. Future work includes scaling action-free pretraining to internet-scale datasets and extending flow-based planning to longer temporal horizons.

REFERENCES

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 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.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

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

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

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*, 2025a.

Xinyi Chen, Yilun Chen, Yanwei Fu, Ning Gao, Jiaya Jia, Weiyang Jin, Hao Li, Yao Mu, Jiangmiao Pang, Yu Qiao, et al. InternVLA-M1: A spatially guided vision-language-action framework for generalist robot policy. *arXiv preprint arXiv:2510.13778*, 2025b.

Yixiang Chen, Peiyan Li, Yan Huang, Jiabing Yang, Kehan Chen, and Liang Wang. Ec-flow: Enabling versatile robotic manipulation from action-unlabeled videos via embodiment-centric flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11958–11968, October 2025c.

Yixiang Chen, Peiyan Li, Jiabing Yang, Keji He, Xiangnan Wu, Yuan Xu, Kai Wang, Jing Liu, Nianfeng Liu, Yan Huang, et al. Bridgev2w: Bridging video generation models to embodied world models via embodiment masks. *arXiv preprint arXiv:2602.03793*, 2026a.

Yuzhi Chen, Ronghan Chen, Dongjie Huo, Yandan Yang, Dekang Qi, Haoyun Liu, Tong Lin, Shuang Zeng, Junjin Xiao, Xinyuan Chang, et al. Abot-physworld: Interactive world foundation model for robotic manipulation with physics alignment. *arXiv preprint arXiv:2603.23376*, 2026b.

Heng Fang, Shangru Li, Shuhan Wang, Xuanyang Xi, Dingkang Liang, and Xiang Bai. Towards generalizable robotic manipulation in dynamic environments. In *European Conference on Computer Vision (ECCV)*, 2026.

Yu Fang, Kanchana Ranasinghe, Le Xue, Honglu Zhou, Juntao Tan, Ran Xu, Shelby Heinecke, Caiming Xiong, Silvio Savarese, Daniel Szafir, et al. Robotic vla benefits from joint learning with motion image diffusion. *arXiv preprint arXiv:2512.18007*, 2025.

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*, 2025a.

Yao Feng, Chendong Xiang, Xinyi Mao, Hengkai Tan, Zuyue Zhang, Shuhe Huang, Kaiwen Zheng, Haitian Liu, Hang Su, and Jun Zhu. Vidarc: Embodied video diffusion model for closed-loop control. *arXiv preprint arXiv:2512.17661*, 2025b.

Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Zhehao Cai, and Lin Shao. FLIP: Flow-centric generative planning as general-purpose manipulation world model. In *The Thirteenth International Conference on Learning Representations*, 2025.

GigaWorld Team, Angyuan Ma, Boyuan Wang, Bohan Li, Chaojun Ni, Guo Li, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, et al. Gigaworld-1: A roadmap to build world models for robot policy evaluation. *arXiv preprint arXiv:2607.02642*, 2026.

Google DeepMind. Introducing Veo 3.1 and advanced capabilities in flow. https://blog.google/innovation-and-ai/products/veo-updates-flow/, October 2025.

Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4d world action modeling from video priors with asynchronous denoising. *arXiv preprint arXiv:2604.26694*, 2026a.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. In *The Fourteenth International Conference on Learning Representations*, 2026b.

Ryan Hoque, Peide Huang, David J Yoon, Mouli Sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. *arXiv preprint arXiv:2505.11709*, 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. In *International Conference on Machine Learning*, pp. 24328–24346. PMLR, 2025.

Yuxin Jiang, Shengcong Chen, Siyuan Huang, Liliang Chen, Pengfei Zhou, Yue Liao, Xindong He, Chiming Liu, Hongsheng Li, Maoqing Yao, et al. Enerverse-ac: Envisioning embodied environments with action condition. *arXiv preprint arXiv:2505.09723*, 2025.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5148–5157, 2021.

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, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In *The Fourteenth International Conference on Learning Representations*, 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*, 2026a.

Peiyan Li, Yixiang Chen, Hongtao Wu, Xiao Ma, Xiangnan Wu, Yan Huang, Liang Wang, Tao Kong, and Tieniu Tan. BridgeVLA: Input-output alignment for efficient 3d manipulation learning with vision-language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Peiyan Li, Yixiang Chen, Yuan Xu, Jiabing Yang, Xiangnan Wu, Jun Guo, Nan Sun, Long Qian, Xinghang Li, Xin Xiao, et al. SpatialVAM: Spatial-aware multi-view video diffusion as a data-efficient robot policy. *arXiv preprint arXiv:2604.03181*, 2026b.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. *arXiv preprint arXiv:2508.05635*, 2025.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Yunfan Lou, Hewen Gao, Xiyu Zhu, Zhuoran Qiao, Xuan Han, Yifan Yang, Yifan Ye, Boxian Yao, and Zhibo Pang. DynamicWAM: Dual-path motion conditioning for world-action models in dynamic manipulation. *arXiv preprint arXiv:2608.00793*, 2026.

Haoxiang Ma, Junhao Cai, Xiaoxu Xu, Hao Li, Yuyin Yang, Yang Tian, Jiafei Cao, Hongrui Zhu, Zherui Qiu, et al. InternVLA-A1.5: Unifying understanding, latent foresight, and action for compositional generalization. *arXiv preprint arXiv:2607.04988*, 2026.

E-Ro Nguyen, Yichi Zhang, Kanchana Ranasinghe, Xiang Li, and Michael S Ryoo. Pixel motion diffusion is what we need for robot control. *arXiv preprint arXiv:2509.22652*, 2025.

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. In *Robotics: Science and Systems (RSS)*, 2025.

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

Kanchana Ranasinghe, Xiang Li, Cristina Mata, Jongwoo Park, and Michael S Ryoo. Pixel motion as universal representation for robot control. *arXiv preprint arXiv:2505.07817*, 2025.

Kanchana Ranasinghe, Honglu Zhou, Yu Fang, Luyu Yang, Le Xue, Ran Xu, Caiming Xiong, Silvio Savarese, Michael S Ryoo, and Juan Carlos Niebles. Future optical flow prediction improves robot control & video generation. *arXiv preprint arXiv:2601.10781*, 2026.

Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, Xin Jin, Ziyou Wang, Lei Jin, Xin Zhang, Yinzhou Tang, Haisheng Su, et al. Worldarena: A unified benchmark for evaluating perception and functional utility of embodied world models. *arXiv preprint arXiv:2602.08971*, 2026.

StarVLA Community. StarVLA: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

Bahey Tharwat, Yara Nasser, Ali Abouzeid, and Ian Reid. Latent action pretraining through world modeling. *arXiv preprint arXiv:2509.18428*, 2025.

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.

Wan Team, Tongyi Lab, Alibaba Group. Wan 2.6. `https://wan.video/introduction/wan2.6`, December 2025.

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 *AAAI Conference on Artificial Intelligence (AAAI)*, 2026.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11097–11107, 2020.

Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. In *Conference on Robot Learning*, pp. 2475–2499. PMLR, 2025.

Lishan Yang, Wenxuan Song, Xi Wang, Pingyue Sheng, Zheng Fang, Ziyang Zhou, Junjie He, Haodong Yan, Jiayi Chen, Nan Sun, Qiao Sun, Lingqiao Liu, Yan Wang, Yuxiang Gao, Feras Dayoub, and Haoang Li. 4D-WAM: Infusing spatiotemporal awareness into world action models through trajectory fields. *arXiv preprint arXiv:2608.08023*, 2026a.

Liudi Yang, Yang Bai, George Eskandar, Fengyi Shen, Mohammad Altillawi, Dong Chen, Ziyuan Liu, and Abhinav Valada. Covar: Co-generation of video and action for robotic manipulation via multi-modal diffusion. *arXiv preprint arXiv:2512.16023*, 2025a.

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

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

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.

Kevin Zhang, Mohit Sharma, Jacky Liang, and Oliver Kroemer. A modular robotic arm control stack for research: Franka-interface and frankapy. *arXiv preprint arXiv:2011.02398*, 2020.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

Zhide Zhong, Haodong Yan, Junfeng Li, Xiangchen Liu, Xin Gong, Tianran Zhang, Wenxuan Song, Jiayi Chen, Xinhu Zheng, Hesheng Wang, et al. Flowvla: Visual chain of thought-based motion reasoning for vision-language-action models. *arXiv preprint arXiv:2508.18269*, 2025.

Shangchen Zhou, Chongyi Li, Kelvin C.K. Chan, and Chen Change Loy. ProPainter: Improving propagation and transformer for video inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

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. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025a.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9834–9844, 2025b.

APPENDIX CONTENTS

A Implementation Details and Training Pipeline **14**

A.1 Architectural Components 14

A.2 Two-Stage Training Strategy 15

A.3 Dataset Construction and Preprocessing 16

A.4 Converting EgoDex to Robot Videos 16

A.5 Input Representation and Flow Encoding 17

A.6 Hyperparameter Summary 17

A.7 Inference Acceleration 17

B Additional Experimental Results **19**

B.1 Per-Task RoboTwin Results 19

B.2 Robustness to Inference-Time Flow Perturbations 23

B.3 Flow Intervention 23

C Benchmark and Evaluation Details **24**

C.1 DOMINO 24

C.2 WorldArena 24

C.3 Real-World Setup 28

D Visualization Results **29**

D.1 RoboTwin Policy Rollouts 29

D.2 Real-World Policy Rollouts 29

D.3 World Model Visualization 29

E Limitations and Broader Impacts **29**

E.1 Limitations 29

E.2 Broader Impacts 29

F AI Use Statement **30**

A IMPLEMENTATION DETAILS AND TRAINING PIPELINE

This section provides comprehensive implementation details to facilitate the reproduction of the training procedure described in Section 3.3 and of the accelerated inference. We detail the architectural components of FlowWAM (Sec. A.1), the two-stage training strategy (Sec. A.2), dataset construction and preprocessing (Sec. A.3), the conversion of EgoDex into robot videos (Sec. A.4), the input representation and flow encoding (Sec. A.5), the hyperparameter configurations for both stages (Sec. A.6), and the inference acceleration (Sec. A.7).

A.1 ARCHITECTURAL COMPONENTS

FlowWAM is constructed by integrating motion-specific modules into a pretrained video generator. The base architecture is Wan2.2-TI2V-5B (Wan et al., 2025), a 5-billion parameter image-to-video

diffusion transformer (DiT) utilizing a UMT5-XXL text encoder and the Wan2.2 causal VAE. The pretrained DiT, VAE, and text encoder architectures are preserved. During training, the VAE and text encoder remain frozen, while the DiT parameters are updated according to the stage-specific protocols detailed below.

Dual-Stream Video Generator. The dual-stream generator incorporates a dedicated pathway for HSV-encoded optical flow while preserving the original video backbone capacity. The flow pathway uses a separate patch embedding layer and output head, both initialized from the corresponding RGB layers. To distinguish the two modalities, a learnable flow-token embedding is added post-patchification as a stream-specific identifier.

Within each self-attention layer, RGB and flow tokens are concatenated for joint attention and subsequently split back into their respective streams. Rotary position embeddings (RoPE) are applied independently within each stream. This design enables both streams to share transformer blocks and latent geometry while preserving distinct identities. Furthermore, because flow frames are compressed with the same VAE as RGB frames, initializing the flow pathway with RGB weights provides a compatible visual prior and significantly accelerates convergence.

Action Expert Architecture. The action expert translates visual-motion latents into low-level robot actions. It is implemented as an AdaLN diffusion transformer ($\sim$780M parameters) comprising 30 layers, matching the depth of the Wan2.2 video DiT. It uses a hidden dimension of 1024, 16 attention heads, and an FFN dimension of 4096. Each expert block contains three components: self-attention over action tokens, cross-attention to visual features (concatenated RGB and flow), and cross-attention to language instructions. Conditioning is further augmented by a proprioceptive token (derived from normalized 14D joint states), which is projected and appended to the T5 instruction context.

Visual Feature Extraction and Training. Rather than directly patchifying VAE latents, the visual features are intermediate hidden representations from the dual-stream video DiT. Specifically, each action-expert block cross-attends to the paired layer of the video DiT via a single additional forward pass. At inference, once the final denoising step completes, the generated RGB and flow latents are passed through the DiT at timestep zero to extract multi-layer hidden states for the action expert. During training, this extraction pass is instead applied to the ground-truth future latents. To improve robustness, these training latents are either kept clean or perturbed with random noise levels provided to the expert (Section 3.2). Finally, the action head predicts the flow-matching velocity of the action chunk, which is integrated numerically during inference.

A.2 Two-Stage Training Strategy

The training pipeline is divided into two stages to decouple general motion representation learning from embodiment-specific action decoding. Both stages utilize the RGB-flow representation and a flow-matching objective, though they differ in supervision, data sources, and trainable parameters.

Stage 1: Action-Free Motion Pretraining. Stage 1 involves pretraining the dual-stream video generator on EgoDex (Hoque et al., 2025), converted to robot videos (Sec. A.4), a dataset of egocentric human manipulation videos without robotic action labels. This stage focuses on learning manipulation-centric motion priors using RGB frames and extracted optical flow. Only the video loss $\mathcal{L}_{\text{video}}$ is optimized, with a learning rate of 5×10^{-5} and updates restricted to the dual-stream DiT.

Stage 2: Joint Embodied Training. Stage 2 initializes the video generator from the Stage 1 checkpoint, integrates the action expert, and optimizes the joint objective (Eq. 5) on RoboTwin 2.0. Input streams from the head, left-wrist, and right-wrist cameras are tiled into a 320×384 configuration (see Sec. A.5). Each chunk generates a rollout of $T_{\text{pixel}} = 9$ pixel frames, from which a 32-step action chunk is decoded (temporal stride 4 between pixel frames). Training is conducted with a learning rate of 1×10^{-4}, using loss weights $\lambda_f = 0.1$ and $\lambda_a = 1.0$, and a motion-aware reweighting strength $\alpha = 2.0$. The trainable parameters include the full DiT, the flow stream, and the action expert.

A.3 Dataset Construction and Preprocessing

RoboTwin Demonstrations. For embodied training, we use RoboTwin 2.0 across 50 bimanual manipulation tasks. Each task contains a *Clean* split with 50 demonstrations under canonical scene configurations and a *Random* split with 500 demonstrations under randomized object poses, distractors, lighting, and backgrounds. Both splits are used during Stage 2 training so that the action expert observes canonical manipulation trajectories as well as visually diverse scene layouts. At evaluation time, success rates are computed over 100 rollouts per task under each setting, following the benchmark protocol.

EgoDex Clip Sampling. For action-free pretraining, we use EgoDex as an unlabeled egocentric manipulation video source. Since EgoDex does not provide robot actions, it is used only for the RGB-flow video objective. We use the converted clips described in Sec. A.4, keeping only the head camera. Frames are resized from 480×270 to 320×256 without cropping, since the rendered arms span the full image width, and each training clip consists of 9 frames sampled with a stride of 4 from the native 30 fps. We chose this stride so that the magnitude distribution of the encoded flow matches that of RoboTwin. Color jitter is applied to RGB frames only, with one draw per clip.

Optical-Flow Extraction. We extract optical flow using RAFT (Teed & Deng, 2020) for both EgoDex and RoboTwin preprocessing. For EgoDex, RAFT is applied to the robot-only frames rendered during conversion (Sec. A.4), so pretraining uses the same robot-only flow target as the robot data. For RoboTwin, we construct robot-only flow targets by replaying the recorded robot joint actions in SAPIEN (Xiang et al., 2020) using the robot URDF and rendering the moving embodiment against a static background. RAFT is then applied to adjacent robot-only frames. This removes background, lighting, and object-motion artifacts from the flow target, yielding cleaner supervision for robot-induced motion. Flow is needed only as a training target in policy mode. At inference, the policy generates future flow itself and requires no rendering. The only exception is the optional history window used in dynamic tasks (Section 4.2), where past flow is rendered from already executed joint states and therefore requires the robot URDF and camera calibration.

HSV Encoding and Magnitude Normalization. The numerical flow field is converted into an HSV image using Eq. 1, where hue represents direction and saturation represents magnitude. We use a 25 px magnitude cap before normalization to prevent large displacements from saturating the color encoding. Displacements below 0.5 px are thresholded to zero to suppress flow noise in nearly static regions.

Multi-View Flow Targets. For Stage 2 RoboTwin training, RGB observations from the head, left-wrist, and right-wrist cameras are tiled into the T-shape input layout. Flow supervision is provided for the head-camera region, where the robot-only replay gives a stable global view of embodiment motion.

A.4 Converting EgoDex to Robot Videos

We convert each EgoDex episode into robot videos in which the human hands are replaced by a rendered robot arm, so that pretraining sees robot embodiments and robot-only flow of the same form as the downstream robot data.

Source data and embodiments. We convert the EgoDex training split (314,839 episodes) with each of 16 robot arms: `aloha`, `aloha_agilex`, `arx_l5`, `fr3`, `gen3`, `iiwa`, `jaco`, `panda`, `piper`, `sawyer`, `ur5e`, `ur10e`, `viperx`, `widowx`, `xarm7`, and `yam`.

Conversion. Hands are segmented with text-prompted SAM 3 (Carion et al., 2025) and removed with ProPainter (Zhou et al., 2023), while objects held in the hand are protected from inpainting. The pipeline also estimates scene depth with Depth Anything 3 (Lin et al., 2025). End-effector targets retargeted from the EgoDex 3D hand tracking are converted into joint configurations by inverse kinematics, and the arm is rendered into the inpainted frames by a camera that follows the head motion. Frames are neither cropped nor stabilized. The same rendering pass also produces robot-only

frames on a black background, from which the flow targets are computed (Sec. A.3), so the flow is aligned with the composited video by construction.

Quality control. Every episode–arm pair carries a certificate with seven flags (self-collision, inter-arm collision, `through_camera`, IK branch switch, unreachable target, `eaten`, and screen occupancy). For pretraining we drop only the pairs whose pixels are corrupted, i.e., those flagged `through_camera` or `eaten`. After filtering, each arm accounts for 4.1–6.9% of the pretraining data, and inverse kinematics is feasible for every arm (median feasibility 100%).

Pretraining. Stage 1 updates only the video DiT and the flow stream: no action expert is constructed. Pretraining uses no robot action labels: the retargeted joint trajectories are derived from the hand poses provided by EgoDex, are used only to render the arm and its robot-only flow, and never enter the loss as action targets. The text prompt names the rendered arm, followed by the EgoDex task instruction. The final run takes 34k steps on 64 A100 GPUs.

A.5 Input Representation and Flow Encoding

Multi-View T-Shape Tiling. To process multi-view inputs without specialized encoders, we tile the head and wrist camera views into a single frame. Following the *T-shape* layout, the head view occupies the top region at its native resolution (320×256), while the left and right wrist views are downsampled and concatenated below it. The resulting 320×384 tile is compatible with the frozen VAE and video generator patch embedding, maintaining architectural consistency across single-view and multi-view scenarios.

Flow Field Generation. The flow stream is designed to capture robot-induced motion. The extraction pipeline follows Sec. A.3: EgoDex flow is computed from the robot-only frames rendered during conversion, while RoboTwin flow is computed from robot-only replay videos to isolate embodiment motion. For the multi-view tile, the head-camera flow occupies the head region and the wrist regions use the constant placeholder.

HSV Encoding and Normalization. The numerical flow field is converted to an HSV image (Eq. 1), mapping direction to hue and magnitude to saturation. We employ the 25 px magnitude cap described in Sec. A.3.

A.6 Hyperparameter Summary

Table 5 summarizes the architectural, data, optimization, and numerical settings used in the two training stages. Pretraining is performed on 64 NVIDIA A100 GPUs, and all downstream training on 32 NVIDIA H100 GPUs across four nodes.

A.7 Inference Acceleration

The accelerated FlowWAM in Table 6 combines few-step distillation of both denoising processes with engineering optimizations that leave the computed actions unchanged. All latencies in Table 6 are measured on a single NVIDIA H100 GPU with matched resolution and prediction horizon.

Video-stream distillation. The dual-stream DiT is distilled from 25 to 4 denoising steps. The 4-step student is the base model with rank-64 LoRA adapters on the attention and feed-forward projections, and the 25-step teacher is the same network with the adapters disabled, so a single copy of the weights serves as both. Teacher and student start from the same noise, and the student is trained so that the action chunk that the frozen action expert decodes from its features matches the chunk decoded from the teacher's features. A small term (weight 1.5×10^{-4}) additionally matches the final RGB and flow latents of the two rollouts. Matching the decoded actions targets the quantity the policy depends on rather than a distance in feature space. The student is trained and sampled with a timestep shift of 4.0, for 2,800 steps at a learning rate of 4×10^{-6} on the RoboTwin training data.

Action-expert distillation. With the distilled video stream fixed, the action expert is distilled from 50 to 10 flow-matching steps. A full-parameter copy of the expert is trained for 3,000 steps

Table 5: Hyperparameters for the two FlowWAM training stages.

Hyperparameter	Value
Video Generator	
Base model	Wan2.2-TI2V-5B
VAE / Text encoder	Wan2.2 VAE / UMT5-XXL (frozen, bf16)
Flow stream init	deep copy of patch embedding & head
Action Expert	
Layers / hidden / heads / FFN	30 / 1024 / 16 / 4096
Parameters	$\sim$780M
Action dim	14
Inputs (RoboTwin, Stage 2)	
Per-camera resolution	320×256
T-shape tiled resolution	320×384
Pixel frames / action steps	9 / 32
Inputs (EgoDex, Stage 1)	
Source	EgoDex converted to 16 robot arms, head camera
Resolution	320×256
Frames per clip	9
Effective fps	7.5 (stride 4 from 30 fps)
Flow	
Estimator	RAFT
Magnitude cap	25 px
Noise threshold	0.5 px
Optimization	
Optimizer	AdamW (weight decay 10^{-2})
Learning rate (Stage 1 / Stage 2)	5×10^{-5} / 1×10^{-4}
Per-GPU batch size (Stage 1 / Stage 2)	1 / 16
Losses & schedules	
Flow weight λ_f	0.1
Action weight λ_a (Stage 2)	1.0
Motion-boost α	2.0
Ref-aug strength	0.1

Table 6: **Inference latency per action chunk**.

Method	Motus (Bi et al., 2025)	FlowWAM w/o flow stream	FlowWAM	FlowWAM distilled
Latency / chunk (ms)	3310	1530	1775	**352**

to reproduce the 50-step teacher's action chunk over the executed part of the chunk, without a ground-truth action term. Because the last step of the action schedule has noise level zero and its output is discarded, the 10-step schedule performs nine network evaluations.

Lossless optimizations. The remaining optimizations only remove redundant computation, and the resulting actions are bit-identical.

(i) All weights stay resident on the GPU instead of being offloaded between chunks.
(ii) The modulation constants are cast once, and the text cross-attention keys and values and the rotary position tables are cached.
(iii) The action expert's cross-attention keys and values over the video features are computed once per chunk instead of at every denoising step.
(iv) One denoising step of the action expert and one of the video DiT are each captured as a CUDA graph and replayed.

B Additional Experimental Results

B.1 Per-Task RoboTwin Results

Tables 7 and 8 report the per-task RoboTwin 2.0 success rates under the in-domain protocol on *Clean* and *Random* scenes, whose 50-task averages appear in Table 1. Tables 9 and 10 report the per-task results under the out-of-domain protocol on *Clean* and on unseen *Random* scenes, respectively.

Table 7: **Per-task RoboTwin 2.0 success rates, in-domain protocol, *Clean* scenes.** Bold: best per row. PT: pretraining.

Simulation Task	π_0	$\pi_{0.5}$	X-VLA	StarVLA	Motus	X-WAM	Fast-WAM	LingBot -VA	FlowWAM w/o PT	FlowWAM w/ PT
Adjust Bottle	99%	**100%**	**100%**	**100%**	89%	**100%**	**100%**	90%	98%	99%
Beat Block Hammer	79%	96%	92%	93%	95%	98%	**99%**	96%	92%	**99%**
Blocks Ranking Rgb	80%	92%	83%	99%	99%	99%	**100%**	99%	99%	**100%**
Blocks Ranking Size	14%	49%	67%	79%	75%	76%	**94%**	**94%**	63%	80%
Click Alarmclock	77%	98%	99%	58%	**100%**	98%	**100%**	99%	97%	99%
Click Bell	71%	99%	**100%**	23%	**100%**	**100%**	**100%**	**100%**	**100%**	99%
Dump Bin Bigbin	88%	92%	79%	91%	95%	90%	**97%**	89%	87%	93%
Grab Roller	98%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	88%	**100%**
Handover Block	47%	66%	73%	97%	86%	88%	95%	**99%**	58%	98%
Handover Mic	97%	98%	0%	98%	78%	88%	**99%**	94%	96%	98%
Hanging Mug	14%	18%	23%	34%	38%	46%	58%	40%	**72%**	64%
Lift Pot	80%	96%	99%	**100%**	96%	**100%**	**100%**	**100%**	91%	**100%**
Move Can Pot	68%	51%	89%	91%	34%	80%	90%	94%	80%	**96%**
Move Pillbottle Pad	67%	84%	73%	98%	93%	96%	**100%**	99%	98%	**100%**
Move Playingcard Away	74%	96%	93%	**100%**	**100%**	**100%**	**100%**	**100%**	98%	**100%**
Move Stapler Pad	41%	56%	78%	74%	83%	67%	77%	91%	68%	**92%**
Open Laptop	71%	90%	93%	**98%**	95%	96%	**98%**	92%	79%	**98%**
Open Microwave	4%	34%	79%	28%	**95%**	89%	62%	82%	61%	78%
Pick Diverse Bottles	69%	81%	58%	87%	90%	**91%**	80%	89%	81%	89%
Pick Dual Bottles	59%	93%	47%	91%	96%	99%	**100%**	**100%**	95%	97%
Place A2b Left	43%	87%	48%	90%	88%	90%	95%	97%	**100%**	97%
Place A2b Right	39%	87%	36%	88%	91%	92%	93%	97%	96%	**98%**
Place Bread Basket	62%	77%	81%	91%	91%	90%	91%	97%	**98%**	94%
Place Bread Skillet	66%	85%	77%	89%	86%	90%	90%	**95%**	87%	92%
Place Burger Fries	81%	94%	94%	**100%**	98%	97%	96%	97%	99%	98%
Place Can Basket	55%	62%	49%	75%	81%	**84%**	71%	81%	71%	66%
Place Cans Plasticbox	63%	94%	97%	**100%**	98%	99%	99%	**100%**	98%	94%
Place Container Plate	97%	**99%**	97%	**99%**	98%	98%	96%	**99%**	95%	**99%**
Place Dual Shoes	59%	75%	79%	91%	93%	83%	**94%**	**94%**	59%	80%
Place Empty Cup	91%	**100%**	**100%**	**100%**	99%	98%	**100%**	**100%**	**100%**	**100%**
Place Fan	66%	87%	80%	94%	91%	84%	96%	**99%**	78%	98%
Place Mouse Pad	20%	60%	70%	87%	66%	84%	83%	93%	**97%**	96%
Place Object Basket	67%	80%	44%	**93%**	81%	85%	89%	91%	77%	90%
Place Object Scale	57%	86%	52%	93%	88%	93%	90%	**96%**	**96%**	94%
Place Object Stand	82%	91%	86%	99%	98%	97%	90%	99%	**100%**	93%
Place Phone Stand	49%	81%	88%	86%	87%	75%	**97%**	**97%**	**97%**	**97%**
Place Shoe	76%	92%	96%	96%	**99%**	97%	96%	98%	96%	**99%**
Press Stapler	44%	87%	92%	**99%**	93%	94%	90%	85%	81%	91%
Put Bottles Dustbin	65%	84%	74%	90%	81%	85%	95%	87%	73%	**97%**
Put Object Cabinet	73%	80%	46%	89%	88%	66%	**94%**	85%	87%	**94%**
Rotate Qrcode	74%	89%	34%	88%	89%	84%	93%	**96%**	69%	**96%**
Scan Object	55%	72%	14%	94%	67%	86%	89%	**96%**	94%	95%
Shake Bottle Horizontally	98%	99%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	97%	**100%**
Shake Bottle	94%	99%	99%	**100%**	**100%**	99%	**100%**	**100%**	**100%**	99%
Stack Blocks Three	72%	91%	6%	94%	91%	97%	95%	**99%**	98%	**99%**
Stack Blocks Two	93%	97%	92%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
Stack Bowls Three	77%	77%	76%	**95%**	79%	88%	80%	86%	80%	82%
Stack Bowls Two	94%	95%	96%	**99%**	98%	98%	92%	94%	96%	98%
Stamp Seal	46%	79%	76%	86%	93%	93%	90%	**96%**	79%	93%
Turn Switch	41%	62%	40%	65%	**84%**	61%	61%	44%	81%	79%
Average (%)	65.92	82.74	72.88	88.18	88.66	89.76	91.88	92.90	87.60	**93.74**

Table 8: **Per-task RoboTwin 2.0 success rates, in-domain protocol, *Random* scenes.** Bold: best per row. PT: pretraining.

Simulation Task	π_0	$\pi_{0.5}$	X-VLA	StarVLA	Motus	X-WAM	Fast-WAM	LingBot -VA	FlowWAM w/o PT	FlowWAM w/ PT
Adjust Bottle	95%	99%	99%	99%	93%	99%	**100%**	94%	97%	99%
Beat Block Hammer	84%	93%	88%	92%	88%	96%	97%	98%	96%	**99%**
Blocks Ranking Rgb	63%	85%	83%	98%	97%	95%	**100%**	98%	97%	98%
Blocks Ranking Size	5%	26%	74%	80%	63%	82%	**98%**	96%	78%	91%
Click Alarmclock	68%	89%	99%	51%	**100%**	99%	**100%**	**100%**	88%	99%
Click Bell	48%	66%	**100%**	27%	**100%**	**100%**	**100%**	**100%**	91%	94%
Dump Bin Bigbin	83%	97%	77%	94%	91%	96%	96%	96%	86%	**98%**
Grab Roller	94%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	98%	**100%**
Handover Block	31%	57%	37%	**93%**	73%	79%	81%	78%	42%	**93%**
Handover Mic	97%	97%	0%	96%	63%	89%	**100%**	96%	98%	**100%**
Hanging Mug	11%	17%	27%	29%	38%	55%	62%	28%	**67%**	61%
Lift Pot	72%	85%	**100%**	**100%**	99%	99%	**100%**	99%	97%	**100%**
Move Can Pot	48%	55%	86%	90%	74%	84%	88%	**97%**	96%	93%
Move Pillbottle Pad	46%	61%	71%	**100%**	96%	98%	99%	99%	**100%**	98%
Move Playingcard Away	65%	84%	98%	98%	96%	98%	**100%**	99%	95%	99%
Move Stapler Pad	24%	42%	73%	90%	85%	70%	64%	79%	72%	**91%**
Open Laptop	81%	96%	**100%**	**100%**	91%	97%	**100%**	94%	88%	98%
Open Microwave	32%	77%	71%	39%	91%	**92%**	45%	86%	68%	88%
Pick Diverse Bottles	31%	71%	36%	86%	91%	**92%**	85%	82%	88%	**92%**
Pick Dual Bottles	37%	63%	36%	93%	90%	**100%**	96%	99%	97%	94%
Place A2b Left	47%	82%	49%	95%	79%	87%	93%	93%	**98%**	94%
Place A2b Right	34%	84%	36%	95%	87%	89%	**99%**	95%	91%	93%
Place Bread Basket	46%	64%	71%	78%	94%	91%	93%	**95%**	89%	94%
Place Bread Skillet	49%	66%	67%	80%	83%	96%	93%	90%	**99%**	90%
Place Burger Fries	76%	87%	94%	**100%**	98%	99%	99%	95%	96%	98%
Place Can Basket	46%	62%	52%	75%	76%	82%	69%	**84%**	68%	62%
Place Cans Plasticbox	45%	84%	98%	**99%**	94%	98%	96%	**99%**	98%	**99%**
Place Container Plate	92%	95%	95%	99%	99%	**100%**	**100%**	97%	92%	**100%**
Place Dual Shoes	51%	75%	88%	**89%**	87%	81%	88%	**89%**	79%	85%
Place Empty Cup	85%	99%	98%	**100%**	98%	99%	**100%**	**100%**	97%	**100%**
Place Fan	71%	85%	75%	95%	87%	92%	96%	93%	81%	**97%**
Place Mouse Pad	20%	39%	70%	94%	68%	86%	89%	**96%**	88%	95%
Place Object Basket	70%	76%	39%	**94%**	87%	87%	88%	88%	62%	88%
Place Object Scale	52%	80%	74%	93%	85%	89%	**97%**	95%	58%	93%
Place Object Stand	68%	85%	88%	98%	97%	96%	94%	96%	**100%**	96%
Place Phone Stand	53%	81%	87%	95%	86%	80%	**99%**	97%	**99%**	**99%**
Place Shoe	76%	93%	95%	**100%**	97%	99%	99%	98%	96%	**100%**
Press Stapler	37%	83%	**98%**	96%	**98%**	90%	97%	82%	70%	91%
Put Bottles Dustbin	56%	79%	77%	85%	79%	95%	90%	91%	**98%**	94%
Put Object Cabinet	60%	79%	48%	**91%**	71%	76%	89%	87%	72%	88%
Rotate Qrcode	70%	87%	33%	90%	73%	83%	89%	91%	90%	**98%**
Scan Object	42%	65%	36%	91%	66%	79%	92%	91%	68%	**96%**
Shake Bottle Horizontally	92%	99%	**100%**	**100%**	98%	99%	**100%**	99%	**100%**	99%
Shake Bottle	91%	97%	**100%**	**100%**	97%	99%	**100%**	97%	99%	97%
Stack Blocks Three	52%	76%	10%	86%	95%	95%	97%	**98%**	87%	96%
Stack Blocks Two	79%	**100%**	87%	**100%**	98%	**100%**	**100%**	98%	97%	**100%**
Stack Bowls Three	75%	71%	86%	**91%**	87%	82%	81%	83%	**91%**	79%
Stack Bowls Two	95%	96%	93%	**100%**	98%	98%	98%	98%	89%	**100%**
Stamp Seal	33%	55%	82%	90%	92%	95%	94%	97%	82%	**98%**
Turn Switch	42%	54%	61%	62%	78%	72%	59%	45%	77%	**80%**
Average (%)	58.40	76.76	72.84	88.32	87.02	90.68	91.78	91.50	87.00	**93.68**

Table 9: **Per-task RoboTwin 2.0 success rates, out-of-domain protocol, *Clean* scenes.** Bold: best per row. PT: pretraining.

Simulation Task	π_0	$\pi_{0.5}$	X-VLA	ABot-M0	StarVLA	X-WAM	Fast-WAM	4D-WAM	FlowWAM w/o PT	FlowWAM w/ PT
Adjust Bottle	90%	99%	99%	99%	84%	92%	99%	80%	97%	**100%**
Beat Block Hammer	43%	93%	87%	72%	75%	92%	82%	92%	97%	**100%**
Blocks Ranking Rgb	19%	72%	3%	61%	39%	85%	81%	88%	98%	**100%**
Blocks Ranking Size	7%	45%	56%	33%	14%	33%	52%	54%	66%	**75%**
Click Alarmclock	63%	65%	62%	0%	60%	95%	99%	**100%**	99%	99%
Click Bell	44%	31%	**100%**	5%	75%	**100%**	95%	98%	98%	97%
Dump Bin Bigbin	83%	92%	83%	89%	54%	88%	94%	86%	**95%**	92%
Grab Roller	96%	99%	**100%**	**100%**	94%	95%	**100%**	**100%**	**100%**	**100%**
Handover Block	45%	30%	77%	16%	6%	21%	66%	79%	94%	**96%**
Handover Mic	98%	98%	98%	90%	66%	97%	98%	85%	**100%**	**100%**
Hanging Mug	11%	17%	19%	7%	1%	17%	14%	**70%**	34%	49%
Lift Pot	84%	98%	96%	51%	68%	77%	91%	88%	99%	**100%**
Move Can Pot	58%	60%	58%	42%	33%	77%	**96%**	94%	89%	93%
Move Pillbottle Pad	21%	46%	66%	56%	40%	78%	92%	89%	95%	**99%**
Move Playingcard Away	53%	84%	84%	78%	62%	83%	94%	82%	**100%**	**100%**
Move Stapler Pad	0%	20%	42%	16%	14%	26%	33%	43%	52%	**71%**
Open Laptop	85%	93%	85%	68%	22%	72%	84%	85%	94%	**97%**
Open Microwave	80%	86%	43%	67%	10%	9%	17%	68%	**95%**	79%
Pick Diverse Bottles	27%	66%	50%	12%	43%	74%	72%	77%	**82%**	**82%**
Pick Dual Bottles	57%	72%	41%	28%	60%	**99%**	87%	92%	96%	97%
Place A2b Left	31%	71%	53%	56%	24%	70%	69%	71%	84%	**89%**
Place A2b Right	27%	68%	49%	41%	25%	59%	60%	62%	82%	**91%**
Place Bread Basket	17%	63%	85%	70%	82%	82%	87%	87%	**94%**	**94%**
Place Bread Skillet	23%	76%	89%	73%	79%	82%	85%	82%	90%	**93%**
Place Burger Fries	80%	82%	90%	93%	85%	93%	97%	86%	93%	**98%**
Place Can Basket	41%	56%	70%	38%	4%	63%	77%	**83%**	72%	53%
Place Cans Plasticbox	34%	31%	93%	56%	95%	93%	**98%**	94%	**98%**	97%
Place Container Plate	88%	92%	94%	**100%**	87%	95%	**100%**	93%	**100%**	98%
Place Dual Shoes	15%	**70%**	30%	28%	30%	38%	67%	66%	44%	69%
Place Empty Cup	37%	96%	90%	97%	85%	93%	90%	**99%**	**99%**	**99%**
Place Fan	20%	66%	63%	48%	30%	47%	59%	48%	75%	**81%**
Place Mouse Pad	7%	34%	53%	31%	11%	49%	55%	54%	78%	**90%**
Place Object Basket	16%	70%	81%	70%	4%	52%	**82%**	75%	78%	**82%**
Place Object Scale	10%	78%	76%	73%	20%	66%	83%	79%	87%	**92%**
Place Object Stand	36%	85%	91%	79%	66%	83%	94%	87%	**96%**	95%
Place Phone Stand	35%	65%	84%	33%	18%	35%	85%	84%	90%	**93%**
Place Shoe	28%	82%	83%	74%	46%	76%	87%	83%	83%	**91%**
Press Stapler	62%	79%	87%	78%	**91%**	85%	90%	76%	84%	85%
Put Bottles Dustbin	54%	78%	47%	46%	11%	71%	55%	68%	79%	**90%**
Put Object Cabinet	68%	39%	41%	41%	29%	28%	42%	**69%**	51%	56%
Rotate Qrcode	68%	90%	81%	52%	11%	47%	81%	79%	78%	**92%**
Scan Object	18%	49%	57%	38%	63%	32%	68%	73%	**90%**	85%
Shake Bottle Horizontally	99%	99%	**100%**	97%	98%	97%	**100%**	98%	**100%**	**100%**
Shake Bottle	97%	**100%**	**100%**	96%	**100%**	96%	**100%**	98%	**100%**	99%
Stack Blocks Three	17%	77%	0%	39%	20%	72%	89%	95%	96%	**98%**
Stack Blocks Two	42%	91%	10%	82%	53%	97%	99%	**100%**	**100%**	**100%**
Stack Bowls Three	66%	81%	77%	76%	46%	69%	80%	82%	**85%**	76%
Stack Bowls Two	91%	94%	85%	92%	65%	93%	94%	94%	**99%**	94%
Stamp Seal	3%	53%	50%	45%	12%	58%	30%	74%	83%	**88%**
Turn Switch	27%	52%	40%	38%	16%	68%	39%	**86%**	62%	68%
Average (%)	46.42	70.66	67.96	57.40	46.52	69.98	77.76	81.50	86.60	**89.24**

Table 10: **Per-task RoboTwin 2.0 success rates, out-of-domain protocol, *Random* scenes.** Bold: best per row. PT: pretraining.

Simulation Task	π_0	$\pi_{0.5}$	X-VLA	ABot-M0	StarVLA	X-WAM	Fast-WAM	4D-WAM	FlowWAM w/o PT	FlowWAM w/ PT
Adjust Bottle	56%	**77%**	33%	49%	0%	13%	0%	36%	26%	62%
Beat Block Hammer	21%	23%	9%	33%	2%	21%	0%	28%	18%	**52%**
Blocks Ranking Rgb	5%	44%	4%	23%	2%	36%	0%	10%	38%	**58%**
Blocks Ranking Size	1%	21%	**25%**	7%	0%	0%	0%	8%	12%	19%
Click Alarmclock	11%	50%	43%	21%	0%	61%	60%	**70%**	57%	59%
Click Bell	3%	35%	63%	11%	7%	52%	7%	77%	**85%**	70%
Dump Bin Bigbin	24%	**84%**	41%	68%	4%	40%	0%	57%	82%	**84%**
Grab Roller	80%	**97%**	20%	91%	44%	16%	2%	68%	59%	88%
Handover Block	8%	12%	1%	0%	0%	21%	0%	25%	17%	**55%**
Handover Mic	13%	6%	1%	22%	1%	8%	0%	13%	17%	**43%**
Hanging Mug	3%	14%	2%	4%	0%	0%	0%	10%	7%	**29%**
Lift Pot	**36%**	35%	15%	29%	0%	0%	0%	9%	3%	21%
Move Can Pot	21%	10%	0%	4%	0%	**34%**	0%	17%	7%	14%
Move Pillbottle Pad	1%	32%	3%	4%	0%	26%	0%	22%	17%	**39%**
Move Playingcard Away	22%	**65%**	5%	50%	0%	11%	0%	35%	12%	21%
Move Stapler Pad	2%	18%	0%	4%	0%	2%	0%	12%	14%	**19%**
Open Laptop	46%	68%	16%	64%	2%	6%	0%	**82%**	52%	75%
Open Microwave	50%	**59%**	5%	26%	1%	24%	0%	12%	20%	43%
Pick Diverse Bottles	6%	42%	30%	7%	15%	26%	0%	44%	42%	**63%**
Pick Dual Bottles	12%	57%	51%	19%	20%	55%	1%	67%	65%	**87%**
Place A2b Left	1%	**49%**	15%	26%	0%	18%	0%	19%	26%	34%
Place A2b Right	6%	39%	13%	22%	1%	21%	1%	42%	32%	**52%**
Place Bread Basket	4%	60%	52%	47%	5%	40%	0%	**77%**	54%	71%
Place Bread Skillet	1%	47%	17%	25%	2%	37%	0%	**69%**	36%	61%
Place Burger Fries	4%	**86%**	35%	55%	11%	36%	0%	72%	74%	84%
Place Can Basket	5%	10%	9%	13%	0%	15%	0%	**51%**	13%	24%
Place Cans Plasticbox	2%	73%	60%	36%	2%	36%	0%	48%	67%	**84%**
Place Container Plate	45%	72%	39%	62%	3%	52%	0%	**82%**	69%	70%
Place Dual Shoes	0%	**45%**	12%	15%	2%	7%	0%	19%	2%	19%
Place Empty Cup	11%	**80%**	16%	43%	0%	36%	0%	70%	66%	75%
Place Fan	10%	26%	1%	10%	1%	6%	0%	17%	5%	**30%**
Place Mouse Pad	1%	**31%**	0%	5%	0%	6%	0%	9%	22%	30%
Place Object Basket	2%	29%	6%	33%	1%	20%	0%	40%	22%	**62%**
Place Object Scale	0%	38%	6%	23%	1%	14%	0%	24%	28%	**41%**
Place Object Stand	11%	47%	43%	25%	0%	28%	0%	42%	56%	**73%**
Place Phone Stand	7%	**37%**	5%	26%	0%	4%	0%	21%	24%	35%
Place Shoe	6%	50%	39%	24%	0%	31%	0%	52%	45%	**71%**
Press Stapler	29%	63%	**73%**	52%	10%	20%	3%	61%	62%	69%
Put Bottles Dustbin	13%	50%	20%	26%	0%	21%	0%	19%	48%	**68%**
Put Object Cabinet	18%	22%	21%	22%	0%	16%	0%	**47%**	8%	17%
Rotate Qrcode	15%	21%	1%	6%	0%	2%	0%	**67%**	4%	12%
Scan Object	1%	40%	9%	17%	5%	14%	0%	**53%**	28%	52%
Shake Bottle Horizontally	51%	**100%**	78%	85%	9%	66%	9%	70%	86%	90%
Shake Bottle	60%	**100%**	76%	84%	5%	67%	10%	72%	86%	92%
Stack Blocks Three	0%	25%	0%	10%	1%	26%	0%	**82%**	28%	54%
Stack Blocks Two	1%	43%	2%	42%	1%	76%	0%	34%	69%	**83%**
Stack Bowls Three	24%	52%	9%	44%	0%	38%	0%	27%	**57%**	49%
Stack Bowls Two	41%	70%	15%	65%	0%	67%	0%	**81%**	75%	74%
Stamp Seal	4%	**21%**	2%	19%	0%	16%	0%	10%	**21%**	19%
Turn Switch	23%	25%	6%	20%	0%	4%	3%	11%	40%	**56%**
Average (%)	16.34	46.00	20.95	30.36	3.16	25.84	1.92	41.80	38.06	**53.04**

B.2 Robustness to Inference-Time Flow Perturbations

Evaluation protocol. We evaluate the robustness of FlowWAM to optical-flow perturbations applied exclusively at inference time, without retraining or adapting the model. We consider Gaussian noise with standard deviations of $\sigma = 2$ and 5 pixels, as well as dropout of 10% of flow pixels. The perturbation is applied according to the operating mode. In policy mode, we perturb the predicted flow used for action decoding, and in world-model mode, we perturb the input flow condition before generating future RGB frames. For policy evaluation, we report the average success rate across the *Clean* and *Random* settings of the RoboTwin 2.0 in-domain protocol. For world-model evaluation, we report EWMScore on the custom validation split used for the world-model ablations.

Table 11: **Robustness to inference-time flow perturbations.** Perturbations are applied to the predicted flow in policy mode and to the input flow condition in world-model mode, without retraining. EWMScore is evaluated on the custom validation split used for the world-model ablations, rather than the official WorldArena evaluation split.

Flow perturbation at inference	Policy success (%) ↑	EWMScore ↑
None	**93.71**	**65.23**
Gaussian noise, $\sigma = 2$ px	93.56	64.45
Gaussian noise, $\sigma = 5$ px	93.41	63.74
Dropout of 10% of flow pixels	93.28	64.03

Results. Table 11 shows that FlowWAM maintains stable performance under the tested perturbations. Across all corruption settings, policy success decreases by at most 0.43 percentage points, while EWMScore decreases by at most 1.49 points. With Gaussian noise of $\sigma = 5$ px, policy success remains at 93.41% and EWMScore at 63.74, compared with 93.71% and 65.23 without perturbation. Similarly, dropping 10% of flow pixels yields a policy success rate of 93.28% and an EWMScore of 64.03. These results indicate that both action prediction and flow-conditioned video generation tolerate the tested levels of synthetic noise and missing flow information.

B.3 Flow Intervention

To test whether the action expert decodes actions from the flow stream rather than relying on the RGB stream alone, we intervene on the flow latents in the feature-extraction pass while keeping the generated RGB latents unchanged. We compare the default setting with three interventions: (i) **Flow from a different task**: the flow latents are replaced with those generated for an episode of an unrelated task; (ii) **Zero-motion flow**: the flow latents are replaced with the encoding of a zero-displacement field; (iii) **Temporally shifted flow**: the predicted flow latents are advanced by one latent frame (4 pixel frames, i.e., 16 control steps), and the last latent frame is repeated. We evaluate on 10 RoboTwin tasks with 100 rollouts each under the in-domain protocol.

Table 12: Flow intervention on 10 RoboTwin tasks (in-domain clean protocol, 100 rollouts per task). The generated RGB latents are identical across rows.

Flow latents given to the action expert	Success (%)
Generated flow (default)	94.2
Flow from a different task	1.2
Zero-motion flow	0.5
Temporally shifted flow (1 latent frame)	36.4

All three interventions reduce success sharply although the RGB latents are unchanged (Table 12). With flow from a different task, the arm moves along trajectories unrelated to the current objects, and success drops to 1.2%. With zero-motion flow, the arm largely remains near its initial pose. Shifting the flow by one latent frame keeps the overall direction of motion toward the target but moves contact-critical events earlier, e.g., the gripper closes before reaching the object, and success drops to 36.4%. The failure follows the direction of the shift, which is difficult to explain as a generic response to out-of-distribution inputs.

Together with Table 11, where i.i.d. Gaussian noise and pixel dropout change success by less than 0.5 points, these results indicate that the action expert is robust to low-level noise in the flow but depends on its motion content: the predicted actions follow the motion plan carried by the flow stream.

C Benchmark and Evaluation Details

C.1 DOMINO

Benchmark. DOMINO (Fang et al., 2026) builds 35 dynamic manipulation tasks on RoboTwin 2.0, in which the targets keep moving, and organizes them into three levels of increasingly complex dynamics. We use the clean Level-1 split with the AgileX platform and evaluate every task over 100 episodes (3,500 in total), reporting the success rate (SR) and the manipulation score (MS), each averaged over tasks.

History window. FlowWAM conditions on four past frames spaced four steps apart plus the current observation, and predicts the next eight frames at the same stride, i.e., 13 pixel frames covering 16 steps of history and 32 steps of future. The Wan2.2 VAE compresses time by a factor of four while keeping the first frame as its own latent, so the five conditioning frames map to exactly two clean latent frames and the eight predicted frames to two further latents. The boundary between clean and denoised latents therefore falls on a latent boundary, and no latent mixes observed with predicted frames. The same window is used for the RGB and the robot-only flow streams. At inference, the history flow is computed from robot-only renderings of the joint states already executed, so it needs no information beyond proprioception.

Training. FlowWAM is trained on the 50 clean Level-1 demonstrations of each task (1,750 episodes in total). The remaining optimization settings follow the RoboTwin configuration in Table 5.

Per-task results. Tables 13 and 14 report the success rate and the manipulation score on each of the 35 tasks.

C.2 WorldArena

This section provides a detailed description of the WorldArena evaluation protocol, including metric definitions and the grouping strategy used to assess visual fidelity, physical consistency, and controllability.

Flow condition. For each episode, we replay the recorded joint trajectory with the robot URDF and the camera parameters provided with the dataset, render robot-only frames, and compute RAFT flow between adjacent frames.

C.2.1 World-Model Training

For world-model training, we uniformly sample 121 frames spanning each complete training trajectory. The first sampled RGB frame serves as the reference observation, and the model learns to generate the remaining frames conditioned on the language instruction and the corresponding robot-only flow sequence. The flow stream remains clean at $t_f = 0$, while noise is applied only to the RGB prediction targets.

C.2.2 Metric Grouping and Aggregate Scoring

The WorldArena benchmark comprises 16 metrics, normalized to a $[0, 100]$ scale where higher values indicate superior performance. The **EWMScore** is defined as the arithmetic mean of these 16 metrics. Detailed results for all metrics are presented in Table 15.

The metrics are organized into six functional groups:

- **Visual Quality**: frame-level realism and feature similarity (IQ, AQ, JEPA).
- **Motion Quality**: the presence and temporal smoothness of motion (DD, FS, MS).

Table 13: **Per-task DOMINO success rates.** Bold: best per row.

Simulation Task	OpenVLA	π_0-FAST	VLA-Adapter	InternVLA-M1	Isaac-GR00T	π_0	OpenVLA-OFT	$\pi_{0.5}$	PUMA	Dynamic WAM	FlowWAM
Adjust Bottle	0%	11%	9%	20%	19%	19%	58%	52%	65%	14%	**72%**
Beat Block Hammer	0%	1%	1%	0%	3%	10%	11%	10%	15%	**84%**	38%
Click Alarmclock	4%	4%	10%	9%	11%	15%	6%	14%	4%	**23%**	16%
Click Bell	1%	0%	1%	2%	2%	0%	3%	0%	3%	0%	**11%**
Dump Bin Bigbin	0%	0%	0%	0%	0%	0%	0%	0%	0%	30%	**65%**
Grab Roller	0%	0%	30%	33%	28%	11%	28%	10%	33%	45%	**85%**
Handover Block	0%	0%	1%	0%	0%	0%	9%	1%	17%	15%	**55%**
Handover Mic	0%	0%	0%	2%	3%	4%	25%	5%	35%	22%	**37%**
Hanging Mug	0%	0%	0%	0%	0%	9%	12%	5%	9%	26%	**28%**
Move Can Pot	0%	9%	0%	0%	1%	16%	29%	7%	22%	11%	**65%**
Move Pillbottle Pad	0%	6%	0%	0%	0%	7%	1%	6%	14%	**91%**	40%
Move Playingcard Away	0%	0%	0%	0%	0%	7%	1%	8%	6%	13%	**40%**
Move Stapler Pad	0%	0%	0%	0%	0%	1%	0%	0%	1%	**39%**	11%
Place A2b Left	0%	1%	0%	0%	0%	2%	1%	2%	13%	8%	**37%**
Place A2b Right	1%	0%	0%	0%	0%	1%	1%	1%	8%	6%	**45%**
Place Bread Basket	0%	2%	1%	0%	1%	6%	1%	8%	12%	**87%**	50%
Place Bread Skillet	0%	0%	13%	12%	10%	2%	1%	9%	19%	9%	**46%**
Place Can Basket	0%	0%	0%	0%	0%	8%	5%	6%	14%	5%	**33%**
Place Container Plate	0%	26%	0%	1%	4%	15%	10%	22%	26%	**77%**	72%
Place Empty Cup	0%	2%	0%	0%	0%	0%	0%	2%	7%	**87%**	50%
Place Fan	0%	0%	0%	0%	2%	2%	1%	2%	8%	**67%**	10%
Place Mouse Pad	0%	0%	0%	0%	0%	4%	1%	5%	2%	**54%**	31%
Place Object Basket	0%	0%	0%	1%	1%	16%	18%	11%	13%	22%	**60%**
Place Object Scale	0%	0%	0%	0%	0%	0%	0%	3%	4%	**36%**	24%
Place Object Stand	0%	2%	0%	0%	1%	1%	0%	1%	1%	**55%**	32%
Place Phone Stand	0%	0%	1%	0%	0%	0%	0%	2%	6%	**80%**	19%
Place Shoe	0%	3%	0%	0%	0%	20%	4%	25%	16%	**91%**	15%
Press Stapler	25%	7%	20%	17%	9%	9%	5%	6%	10%	**53%**	43%
Put Bottles Dustbin	1%	3%	16%	18%	13%	20%	19%	17%	23%	24%	**43%**
Put Object Cabinet	0%	1%	2%	7%	14%	11%	1%	11%	**34%**	7%	15%
Rotate Qrcode	0%	0%	0%	0%	0%	6%	11%	4%	**14%**	11%	12%
Scan Object	0%	0%	0%	0%	3%	0%	4%	1%	5%	1%	**19%**
Shake Bottle	14%	18%	27%	39%	43%	28%	17%	33%	55%	33%	**88%**
Shake Bottle Horizontally	8%	26%	23%	28%	46%	33%	31%	42%	75%	43%	**88%**
Stamp Seal	0%	2%	0%	0%	0%	3%	3%	6%	13%	**68%**	51%
Average (%)	1.54	3.54	4.43	5.40	6.11	8.17	9.06	9.63	17.20	38.20	**41.31**

- **Content Consistency**: stability of subjects, backgrounds, and photometric properties (SC, BC, PC).
- **Physics Adherence**: interaction plausibility and trajectory accuracy (IntQ, TA).
- **3D Accuracy**: depth consistency and perspectivity (DA, Per).
- **Controllability**: uniformity with instructions and action-conditioned variations (InsF, SA, AcF).

C.2.3 Metric Definitions and Implementation

We follow the official WorldArena metric definitions (Shang et al., 2026). Each raw metric is normalized to the common scoring range before computing EWMScore, so the descriptions below focus on what each metric measures and the main evaluator used by WorldArena.

Visual Quality.

- **IQ** (Image Quality) assesses the technical quality of individual frames using MUSIQ (Ke et al., 2021). It targets no-reference distortions such as blur, overexposure, sensor noise, and compression artifacts, and averages the per-frame quality estimates over the video.
- **AQ** (Aesthetic Quality) evaluates whether frames have visually pleasing lighting, color composition, and perceptual appeal. WorldArena computes this score with the LAION aesthetic predictor, which maps each frame into an aesthetic feature space and averages the predicted frame scores.
- **JEPA** (JEPA Similarity) measures high-level spatiotemporal similarity between generated and reference videos. WorldArena extracts features with a pretrained V-JEPA encoder (Bardes et al., 2024) and computes a maximum mean discrepancy (MMD) score with a second-order

Table 14: **Per-task DOMINO manipulation scores.** Bold: best per row.

Simulation Task	OpenVLA	π_0-FAST	VLA-Adapter	InternVLA-M1	Isaac-GR00T	π_0	OpenVLA-OFT	$\pi_{0.5}$	PUMA	Dynamic WAM	FlowWAM
Adjust Bottle	1.73	37.09	38.07	62.02	50.82	46.89	63.57	71.45	74.04	50.30	**82.70**
Beat Block Hammer	4.23	18.13	15.71	17.18	21.53	23.30	23.20	23.22	29.09	**86.70**	50.00
Click Alarmclock	6.02	10.63	17.00	17.12	18.50	18.95	10.53	18.47	8.38	**25.40**	20.40
Click Bell	3.29	9.32	11.56	15.30	14.82	8.04	11.24	6.35	13.64	11.70	**22.60**
Dump Bin Bigbin	2.43	13.37	23.93	30.52	26.91	20.88	17.84	16.64	39.05	34.90	**76.20**
Grab Roller	3.21	15.59	59.19	60.47	53.56	31.46	43.87	34.32	56.48	68.20	**90.20**
Handover Block	7.02	25.35	54.15	38.71	44.88	39.21	61.17	33.95	64.79	48.80	**75.70**
Handover Mic	2.34	21.61	19.06	33.10	35.52	26.18	37.58	29.10	54.40	39.70	**57.20**
Hanging Mug	3.58	13.39	30.31	28.51	32.21	26.03	35.49	25.59	41.74	40.20	**51.50**
Move Can Pot	6.80	49.88	24.25	26.55	35.09	41.72	57.26	40.34	47.80	46.40	**75.70**
Move Pillbottle Pad	1.86	19.19	13.24	15.41	12.94	16.53	9.93	15.93	21.82	**91.40**	46.80
Move Playingcard Away	3.26	20.00	22.26	24.19	25.95	21.53	7.94	24.84	27.90	38.20	**57.90**
Move Stapler Pad	3.54	17.12	13.46	14.33	15.97	11.96	10.87	13.38	16.24	**47.10**	25.00
Place A2b Left	3.10	28.69	23.02	25.63	27.02	21.81	22.37	26.04	41.65	31.90	**58.50**
Place A2b Right	6.14	27.23	22.11	27.24	26.63	20.08	25.00	23.63	34.39	33.60	**65.60**
Place Bread Basket	4.45	17.95	15.04	13.30	18.65	11.67	7.98	15.89	22.44	**88.50**	56.50
Place Bread Skillet	7.31	13.39	35.87	35.34	30.34	16.11	20.32	22.30	35.95	35.30	**61.00**
Place Can Basket	12.21	24.49	21.33	21.14	30.30	37.49	37.78	29.48	45.79	39.80	**65.00**
Place Container Plate	4.70	35.47	7.50	16.56	19.55	21.79	16.87	28.46	34.45	**79.00**	75.30
Place Empty Cup	2.98	12.11	8.41	14.37	15.68	8.05	7.74	12.71	17.85	**88.50**	56.60
Place Fan	6.04	10.93	15.14	12.14	11.88	9.91	6.25	11.72	14.36	**69.90**	18.70
Place Mouse Pad	3.08	18.23	13.74	15.15	17.91	14.93	13.61	19.03	17.86	**61.30**	41.50
Place Object Basket	6.88	25.51	23.10	26.16	31.11	33.55	37.36	30.19	41.67	42.10	**76.40**
Place Object Scale	3.92	15.18	14.31	14.61	16.87	9.71	10.92	15.37	16.05	**42.60**	35.50
Place Object Stand	7.55	11.97	12.40	17.41	17.65	8.00	8.53	8.97	10.66	**58.40**	39.00
Place Phone Stand	3.16	19.25	14.46	16.26	18.84	13.11	12.66	17.98	19.02	**84.60**	35.10
Place Shoe	4.45	16.13	12.53	17.87	17.21	32.11	17.39	32.84	27.23	**92.80**	31.20
Press Stapler	27.12	16.30	29.49	28.98	22.64	18.77	13.53	13.27	18.40	**60.20**	52.00
Put Bottles Dustbin	7.25	18.76	34.84	39.65	33.42	37.01	29.81	33.30	36.82	41.20	**61.20**
Put Object Cabinet	6.54	24.13	38.14	47.28	50.22	41.38	42.07	46.24	**61.64**	46.50	58.00
Rotate Qrcode	3.67	16.75	19.19	25.05	28.08	15.90	23.31	17.90	**29.27**	21.20	28.80
Scan Object	9.69	13.77	27.82	33.85	30.69	19.64	19.93	20.48	30.58	29.70	**41.80**
Shake Bottle	18.49	30.88	51.16	58.41	60.65	44.02	23.83	49.40	65.37	49.20	**91.60**
Shake Bottle Horizontally	12.33	41.11	51.98	56.48	67.33	53.03	38.75	62.72	80.57	62.20	**93.00**
Stamp Seal	3.13	21.38	17.10	18.82	19.65	17.75	15.47	22.58	26.50	**73.10**	60.00
Average	6.10	20.87	24.31	27.57	28.60	23.96	24.06	26.12	34.97	53.16	**55.26**

Table 15: **Per-metric WorldArena results.** All scores are reported on a $[0, 100]$ scale.

	Visual Quality			Motion Quality			Content Consistency			Physics		3D Accuracy		Controllability			
Method	IQ	AQ	JEPA	DD	FS	MS	SC	BC	PC	IntQ	TA	DA	Per	InsF	SA	AcF	**EWMScore**
CogVideoX	36.23	36.30	94.84	31.47	21.77	73.30	80.97	88.38	33.79	66.96	34.79	91.09	85.46	75.58	89.70	0.00	58.79
Veo 3.1	65.57	**48.79**	57.97	14.66	8.26	67.85	75.82	**91.67**	37.52	81.50	11.36	74.27	**99.64**	**97.14**	83.79	8.52	57.77
Wan 2.2	39.53	38.32	72.02	21.46	10.91	67.07	72.91	79.37	**47.57**	54.54	16.93	80.57	81.98	49.84	88.98	5.39	51.71
Wan 2.6	**67.36**	44.40	72.57	33.63	22.01	**82.12**	73.15	84.29	34.57	73.16	12.18	75.43	93.94	89.96	88.09	9.92	59.80
Cosmos-Predict 2.5 (text)	65.49	44.33	46.69	21.62	17.55	70.93	66.19	73.98	35.39	65.14	11.60	70.71	98.22	64.70	86.34	**10.01**	53.06
Cosmos-Predict 2.5 (action)	49.48	34.14	90.65	18.96	9.95	65.66	67.24	72.72	45.21	60.46	27.49	90.35	84.68	60.54	89.86	1.31	54.29
IRASim	41.85	32.86	93.72	21.60	11.66	64.31	72.82	78.69	45.70	62.76	35.92	93.50	84.16	65.40	89.43	3.94	56.15
Vidar	41.75	39.87	61.10	21.05	17.03	79.11	70.37	78.24	20.57	60.84	21.26	79.77	81.88	61.54	88.46	7.86	51.92
Genie Envisioner	28.18	26.39	33.81	32.41	19.91	69.17	74.53	87.54	16.04	22.74	2.63	86.83	54.32	20.36	85.98	1.07	41.37
Ctrl-World	42.44	37.05	92.77	**41.82**	**33.57**	77.34	**83.56**	90.30	12.88	62.62	48.20	93.25	83.66	67.68	88.68	3.90	59.98
ABot-PhysWorld (text)	61.03	41.83	90.36	39.33	29.22	76.48	80.56	89.02	20.52	**81.96**	31.50	71.99	98.94	92.10	89.58	7.65	62.63
GigaWorld-1	51.18	41.17	96.77	30.52	18.64	68.33	80.97	86.43	28.11	75.10	54.27	98.44	95.60	82.14	89.42	0.28	62.34
FlowWAM (Ours)	48.87	40.83	**97.14**	30.41	20.47	71.65	82.46	89.97	29.78	73.82	**64.26**	**98.97**	94.56	82.76	**89.93**	3.50	**63.71**

polynomial kernel, so higher values indicate that the generated video distribution is closer to the ground-truth demonstration distribution.

Motion Quality.

- **DD** (Dynamic Degree) quantifies the strength of salient motion in the video. WorldArena estimates RAFT (Teed & Deng, 2020) optical flow between adjacent frames, focuses on the top 5% active pixels, and maps their average magnitude through a resolution-adaptive scoring function.
- **FS** (Flow Score) measures the overall motion intensity across the full video. It averages dense RAFT flow magnitudes over pixels and time, then normalizes the raw value so that videos with meaningful dynamic interaction receive higher scores.

- **MS** (Motion Smoothness) evaluates temporal coherence under motion. WorldArena uses a video frame interpolation model to reconstruct intermediate frames from neighboring frames, compares the interpolation with the real middle frame using SSIM, and weights the score by motion magnitude to avoid over-rewarding static videos.

Content Consistency.

- **SC** (Subject Consistency) measures whether foreground objects and robot parts remain semantically stable across frames. It compares DINO feature similarity between the current frame, the first frame, and the previous frame, with a dynamic-degree penalty so that nearly static videos cannot obtain artificially high consistency scores.
- **BC** (Background Consistency) assesses whether the scene background remains stable over time. It uses CLIP image features to compare each frame with the first and previous frames, and applies the same dynamic-degree adjustment to avoid rewarding static or frozen rollouts.
- **PC** (Photometric Consistency) evaluates pixel-level texture stability during motion. WorldArena warps frames forward and backward with optical flow, computes the average endpoint reconstruction error, and converts the reciprocal error into a normalized score with a dynamic-degree penalty.

Physics and 3D Accuracy.

- **IntQ** (Interaction Quality) measures whether robot-object interactions obey plausible physical behavior. WorldArena uses Qwen3-VL as a VLM judge to score visible contact, force transfer, friction, penetration, and object response on a 1–5 Likert scale before normalization.
- **TA** (Trajectory Accuracy) quantifies whether the generated robot-arm trajectory follows the ground-truth execution. It extracts robotic-arm bounding boxes with SAM3 (Carion et al., 2025), interpolates missing detections when necessary, and computes a normalized dynamic time warping (NDTW) alignment score between generated and reference trajectories.
- **DA** (Depth Accuracy) evaluates whether the generated rollout preserves the scene's spatial geometry. WorldArena estimates monocular depth for generated and reference frames, applies median-based scale alignment to handle depth-scale ambiguity, computes absolute relative depth error, and inverts the normalized error so that higher values indicate better geometry.
- **Per** (Perspectivity) assesses 3D plausibility beyond depth-map matching. A Qwen3-VL-based judge evaluates perspective cues such as scale changes with depth, occlusion relationships, lighting consistency, and camera-geometry stability on a normalized Likert scale.

Controllability.

- **InsF** (Instruction Following) evaluates whether the generated video follows the requested action type, target object, and final task state. WorldArena uses Qwen3-VL as a VLM judge with a normalized 1–5 scale, and the rubric penalizes failures such as wrong objects, wrong manipulation type, or hallucinated human hands in place of robot arms.
- **SA** (Semantic Alignment) measures semantic agreement between generated and reference executions. WorldArena first uses Qwen2.5-VL to produce structured descriptions of both videos, then computes CLIP text-feature similarity between those descriptions.
- **AcF** (Action Following) evaluates whether a model responds differently to different action instructions instead of collapsing to repetitive rollouts. For a shared initial frame, WorldArena generates multiple distinct action instructions, extracts global CLIP features from the resulting videos, and scores the average pairwise feature dissimilarity.

C.2.4 Performance Analysis on WorldArena

The comprehensive metric profile in Table 15 reveals that FlowWAM's performance gains are primarily driven by metrics requiring action-grounded prediction. Specifically, FlowWAM achieves state-of-the-art results in **Trajectory Accuracy**, **Depth Accuracy**, and **Semantic Alignment**. These results suggest that flow conditioning provides a spatially aligned signal that guides the video generator in accurately modeling robot dynamics, rather than merely improving generic image aesthetics. While FlowWAM maintains competitive performance in subject and background consistency, its core strength lies in its ability to generate physically plausible and reliable motion rollouts.

C.3 Real-World Setup

This section provides additional details regarding the real-world experimental setup, including the robot platforms, camera configurations, and the task suite used for evaluation.

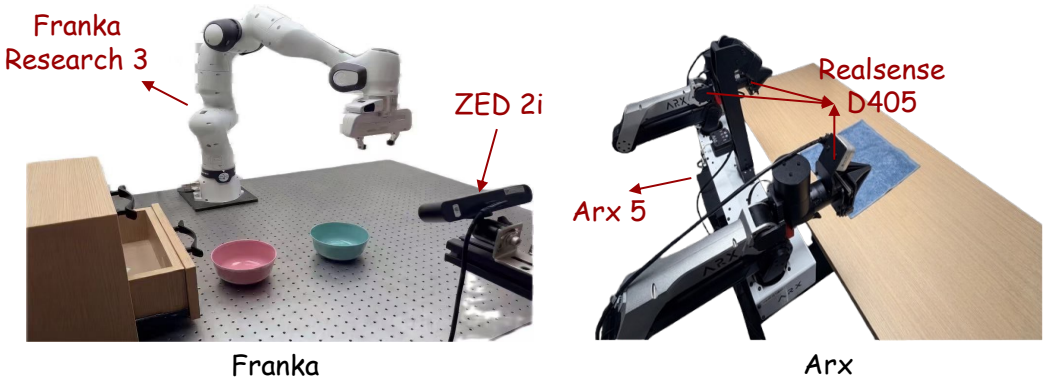

Figure 4: **Real-world experimental setup.** We evaluate FlowWAM on two physical platforms: a single-arm Franka Research 3 robot and a bimanual ARX 5 robot.

C.3.1 Robot Platforms and Sensors

We evaluate FlowWAM on two robot platforms:

- **Franka Research 3**: a single-arm manipulation platform controlled through FrankaPy (Zhang et al., 2020), equipped with a parallel-jaw gripper and one ZED camera for workspace observation, and used for precision pick-and-place and contact-rich object manipulation.
- **ARX 5**: a bimanual robot platform using an OCS2-based optimal-control stack, equipped with three Intel RealSense D405 cameras, including one front camera for the global workspace view and two wrist cameras for close-range manipulation, and used for tasks that require coordinated motion between two arms.

C.3.2 Task Suite and Data Collection

The real-world evaluation suite consists of seven tasks designed to test spatial precision, object interaction, and bimanual coordination:

- **Stack Bowls** (Franka): pick up one bowl and place it stably on top of another bowl.
- **Place in Drawer** (Franka): grasp an object and insert it into an open drawer while avoiding collisions with the drawer boundary.
- **Put in Plate** (Franka): move a target object into a plate, requiring accurate placement relative to a shallow container.
- **Place Two Cups** (Franka): sequentially manipulate two cups and place them at the desired target locations.
- **Fold Towel** (ARX): use both arms to grasp and fold a deformable towel.
- **Stack Bowls (II)** (ARX): coordinate two arms to grasp, transfer, and stack bowls in a stable configuration.
- **Clean Plate** (ARX): execute a bimanual cleaning motion over a plate, requiring coordinated arm motion over an extended horizon.

For each task, all methods are trained on the same 100 teleoperated demonstrations and evaluated over 50 trials with randomized object poses. Flow targets are computed as for RoboTwin (Appendix A.3), from robot-only frames rendered with each robot's URDF and calibrated camera, and are used only during training.

D VISUALIZATION RESULTS

This appendix provides qualitative visualizations that complement the quantitative evaluations in Section 4. The goal is to make the behavior of FlowWAM directly inspectable. Beyond aggregate metrics, the generated rollouts should preserve the intended robot motion, object interaction, and scene geometry over time.

D.1 ROBOTWIN POLICY ROLLOUTS

Figures 5 and 6 show policy-mode rollouts on a broad set of RoboTwin 2.0 (Chen et al., 2025a) tasks under both the *Clean* and *Random* settings. For each task, we show the predicted RGB rollout together with its corresponding optical-flow plan, and the executed simulation rollout decoded by the action expert. The *Clean* examples (Figure 5) cover diverse bimanual manipulation skills under the canonical layout, while the *Random* examples (Figure 6) probe the same skills under randomized object pose, distractor placement, lighting, and background. Across both settings, the predicted flow concentrates on the moving embodiment and traces a coherent goal-directed trajectory, while the RGB rollout preserves the surrounding object context. The decoded actions follow this robot-motion plan to complete the task, and the qualitative behavior remains stable as the scene is randomized, mirroring the quantitative robustness reported in Tables 7 and 8.

D.2 REAL-WORLD POLICY ROLLOUTS

Figure 7 visualizes FlowWAM's policy mode behavior on the real-world manipulation suite introduced in Section 4.4, covering both the single-arm Franka platform and the bimanual ARX platform. The first four rows show single-arm Franka tasks (from top to bottom, *Place Two Cups*, *Place in Drawer*, *Put in Plate*, *Stack Bowls*), which stress spatial precision and object contact. The last three rows show dual-arm ARX tasks (*Clean Plate*, *Fold Towel*, *Stack Bowls (II)*), which additionally require bimanual coordination and longer-horizon motion. For each task, we juxtapose the model-side prediction (predicted RGB rollout and the corresponding optical-flow plan) with the realized robot execution after the predicted flow is decoded into continuous actions by the action expert. This side-by-side layout makes the role of flow directly inspectable across both single-arm and bimanual settings: the predicted flow provides a dense motion plan in image space, while the executed rollout shows whether that plan translates into the intended end-effector motion and object interaction on the physical robot.

D.3 WORLD MODEL VISUALIZATION

Figure 8 visualizes representative action-conditioned world-model predictions. Given the initial observation and the target action condition, FlowWAM generates future RGB frames together with the corresponding motion structure. The visualization illustrates how the dense optical-flow representation provides spatially grounded guidance for video prediction, helping the model maintain coherent robot-object motion and reduce action drift across the rollout.

E LIMITATIONS AND BROADER IMPACTS

E.1 LIMITATIONS

FlowWAM represents motion as 2D image-space displacement, which does not directly encode contact forces or large out-of-plane motion. Extending the representation to 3D is a natural direction for future work. Furthermore, our action-free pretraining relies on the 3D hand poses provided by EgoDex. Extending it to in-the-wild videos would require monocular hand-pose and camera estimation, which we have not yet studied.

E.2 BROADER IMPACTS

FlowWAM studies flow-based action representations for embodied world models in simulated and lab-scale manipulation, with potential benefits for more reusable and data-efficient robot learning. We do not anticipate direct societal impacts beyond those generally associated with foundational robot-learning research.

F AI USE STATEMENT

We used generative AI tools only for general-purpose writing. Specifically, large language models were used to polish the grammar and wording of author-written text. The authors take full responsibility for the final content of this work, including all text, claims, and artifacts.



Figure 5: Qualitative policy rollouts on RoboTwin 2.0 *Clean* tasks. For each task, the small panels show FlowWAM's predicted RGB frames and the corresponding optical-flow plan, and the large panels show the simulation rollout after the action expert decodes the predicted flow into actions.

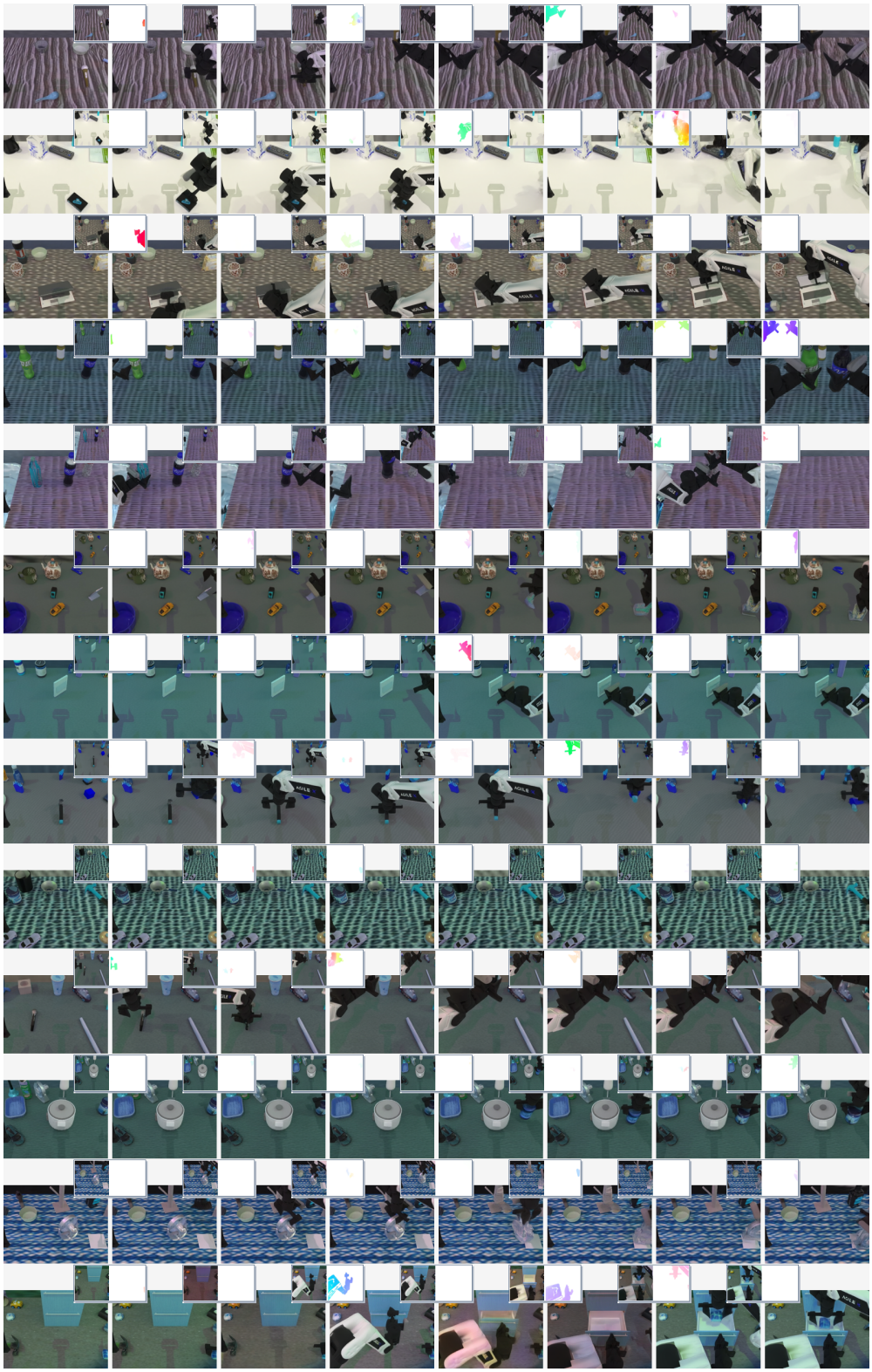

Figure 6: Qualitative policy rollouts on RoboTwin 2.0 *Random* tasks. Compared with the *Clean* setting in Figure 5, the scenes here include randomized object pose, distractors, lighting, and background.



Figure 7: Real-world execution examples in policy mode across single-arm and dual-arm platforms. The first four rows correspond to single-arm Franka tasks, and the last three rows correspond to dual-arm ARX tasks. Within each task, FlowWAM's predicted RGB frames and optical-flow plans are paired with the corresponding physical executions. The qualitative alignment between the predicted motion fields and the realized robot trajectories shows how the video-native flow representation connects model prediction with executable control.



Figure 8: Qualitative visualization of FlowWAM's action-conditioned world model. The examples show how the predicted future frames follow the requested manipulation dynamics while preserving object layout and scene consistency.