WHAT LIMITS CROSS-TASK GENERALIZATION IN WORLD ACTION MODELS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-task generalization remains difficult for world action models (WAMs), even when their predicted futures depict plausible manipulation. This gap can be attributed directly: we make this gap directly testable by decoupling a WAM into a world model and an inverse-dynamics model (IDM), then holding the IDM fixed while replacing the source of future images. We introduce *Align Once, Evaluate Everywhere* (AoE), a camera-space inverse-dynamics model pretrained on large-scale robot data, which provides this recovery interface for different future sources across tasks and environments. Through it, we find that actions recovered from recorded futures of unseen tasks execute almost as well as replayed demonstrations, whereas futures predicted by world models collapse across benchmarks. The failures are not generic: they concentrate on the gripper and interaction geometry while generation capacity is spent on task-irrelevant background. This diagnosis suggests that supervision should target exactly these regions. We therefore introduce the *vision-to-action* (VITA) loss, which passes generated futures through the frozen AoE and penalizes discrepancies in the recovered actions, redirecting generation toward action-relevant detail. Across two world models and three manipulation benchmarks, this simple change raises the execution success of generated futures from near zero to 47.0%. The same loss also transfers to WAMs and yields consistent gains in native control.

1 INTRODUCTION

World action models (WAMs) offer a compelling premise for embodied tasks: use the generalization capabilities of world models (WMs) to imagine future scenes, then condition an action head on the latent representations of those imagined futures to generate actions (Li et al., 2026a; Yuan et al., 2026). This approach achieves strong performance on in-distribution tasks and under modest distribution shifts, yet falls substantially behind in more challenging settings, such as generalization across environments or tasks. This gap is often attributed to the difficulty of predicting 3D poses or to action heads that are tightly coupled to the environment encountered during training. Recent studies have also identified problems in world model predictions of robotic scenes, including physically implausible behavior and 2D-like flat appearance (Shang et al., 2026; Co et al., 2026). However, these analyses remain confined to 2D future videos that cannot be executed directly, leaving the impact of world model predictions on downstream cross-task generalization unquantified.

Evaluating these videos through an inverse dynamics model (IDM) can also be inconclusive: exposure to trajectories from downstream environments during training may allow the IDM to compensate for distorted predictions, obscuring whether the predicted futures themselves support execution. To better evaluate the executability of future videos predicted by world models, we develop *Align Once, Evaluate Everywhere* (AoE). Pretrained on 9,637h of robot data, AoE uses camera-space representation to decouple action prediction from downstream environments. With little data from a downstream environment, AoE translates 2D gripper displacements and depth changes into executable poses. This enables interactive execution beyond the environments seen during pretraining, while keeping action trajectories out of training. On the same unseen benchmarks, we observe a substantial gap between recorded videos and world model predictions: actions recovered from recorded videos approach the success rate of replaying recorded actions, whereas those recovered from predicted future videos perform poorly. 2D flat appearance, physically implausible pose, and hallucinated scene substantially degrade the quality of the recovered poses (Liu et al., 2026a; Mi et al., 2026).

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

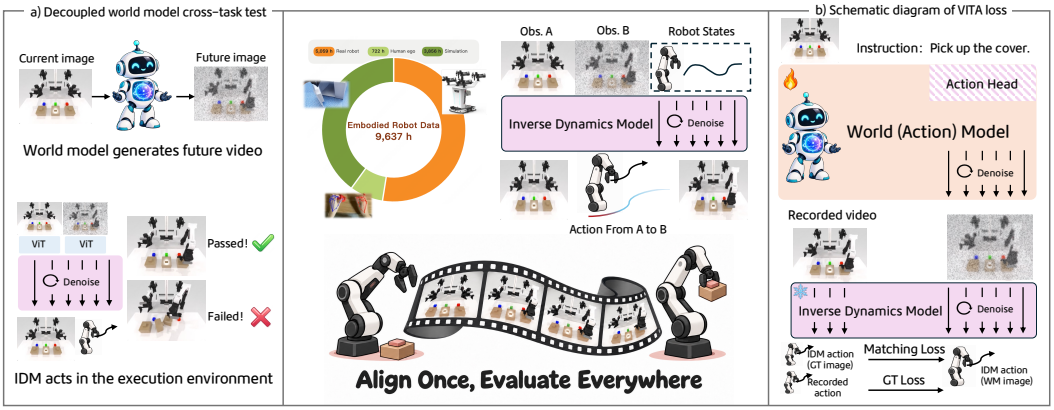


Figure 1: Overview of Align Once, Evaluate Everywhere (AoE). Trained on 9,637 hours of robot image–action pairs, AoE infers the action between any two frames, serving as a bridge from video to action. *Left:* AoE enables evaluation of WMs: dreamed task completions in unseen environments are decoded into executable actions and assessed in the real environment. *Right:* The proposed vision-to-action (VITA) loss further improves the world (action) model by aligning actions recorded and inferred from dreamed and recorded videos.

To address this limitation, we introduce the *vision-to-action* (VITA) loss, which uses AoE as a frozen supervisor to guide world models toward preserving the visual details needed for physically accurate gripper motion. We pass predicted future videos through AoE and use discrepancies of action-space representation to supervise action-relevant visual details (Wang et al., 2026; Feng et al., 2025). AoE's sensitivity to gripper motion directs this supervision toward regions that matter for downstream execution (Wi et al., 2023; Wu et al., 2020). Gradients propagate through AoE to the world model while AoE itself remains frozen. VITA is natively suitable for world action model training to supervise predicted visual futures, allowing action-grounded visual supervision to shape the representations used for native action prediction. This supervision is applied during training and does not require the model to explicitly generate future images during execution.

Our experiments examine three aspects of this approach. First, we use AoE to evaluate the cross-task executability of future videos generated by existing world models. Comparing generated futures with recorded videos reveals a substantial execution gap, highlighting a limitation in the transfer of world-model predictions to unseen tasks. Second, we evaluate VITA in both world models and world action models. VITA substantially improves the execution success of actions recovered from generated futures and improves native action prediction in LingBot-VA (Li et al., 2026a) and FastWAM (Yuan et al., 2026), with particularly pronounced gains under distribution shifts (Zheng et al., 2026; Zhen et al., 2026). FastWAM benefits even though it does not explicitly generate future images during inference. Third, we study how scaling AoE's pretraining data affects its roles as both an evaluator and a supervisor. Larger-scale pretraining improves action recovery on unseen tasks and yields stronger downstream performance when the frozen AoE is used for VITA supervision.

Our contributions are threefold: (1) we introduce AoE, an inverse dynamics model that uses the camera-space action representation to support zero-shot cross-task generalization; (2) we propose VITA, an action-grounded loss that uses frozen AoE supervision to guide both WMs and WAMs to preserve visual details needed for executable 3D pose changes; and (3) we demonstrate that VITA raises the cross-task execution success of world-model-generated futures from near zero to 47.0% and improves native action generation in WAMs under both in-distribution and out-of-distribution.

2 Preliminaries

Forward and inverse dynamics. Let o_t denote the robot's observation at time t, a_t its action, and ℓ a language instruction specifying the task. A forward dynamics model (FDM) predicts the consequences of an action. Written in observation space, its prediction takes the form

$$\hat{o}_{t+1} = \mathcal{F}_\theta(o_t, a_t), \tag{1}$$

where θ denotes the model parameters and $\hat{o}_{t+1}$ is the predicted next observation. An inverse dynamics model (IDM) infers an action consistent with an observed transition:

$$\hat{a}_t = \mathcal{D}_\phi(o_t, o_{t+1}), \tag{2}$$

where ϕ denotes the IDM parameters. These two mappings connect actions and observations in complementary directions: forward dynamics predicts what an action will produce, while inverse dynamics recovers an action that could produce a given transition. Applied to selected frame pairs from a video, inverse dynamics provides a way to translate visual changes into robot actions.

From world models to world action models. In embodied tasks, a world model provides the visual-prediction component of a policy. An instruction-conditioned world model predicts a task-directed future observation from the current observation and instruction. WAMs integrate this prediction process with action generation (Li et al., 2026a; Yuan et al., 2026). In attention-based architectures, an action head reads internal visual features associated with the predicted future. We summarize this interface as

$$h_t = \mathcal{H}_\theta(o_t, \ell), \qquad \hat{a}_t = \mathcal{A}_\psi(o_t, \ell, h_t), \tag{3}$$

where h_t denotes the latent features associated with future prediction, $\mathcal{H}_\theta$ denotes their computation within the world model, and $\mathcal{A}_\psi$ is the action head. Attention allows the action head to access h_t during action prediction. Training this coupled system on downstream data ties the visual features to the action head that consumes them. Future imagination therefore influences control through an internal representation, and task success reflects the behavior of the combined system. This makes it difficult to directly assess how the predicted visual future itself supports execution on unseen tasks.

We introduce an explicit image-space interface between future prediction and action recovery:

$$\hat{o}_{t+1} = \mathcal{W}_\theta(o_t, \ell), \qquad \hat{a}_t = \mathcal{D}_\phi(o_t, \hat{o}_{t+1}), \tag{4}$$

where $\mathcal{W}_\theta$ is the instruction-conditioned world model and $\mathcal{D}_\phi$ is an independently pretrained IDM. We first evaluate the IDM on recorded transitions from unseen tasks. We then replace the recorded future observations with world-model predictions while keeping the IDM frozen. This shared action-recovery interface allows us to evaluate IDM transfer and the executability of generated futures separately under the same execution protocol.

3 METHOD

We introduce AoE, an inverse dynamics model pretrained on 9,637 hours of robot image–action pairs, and use it as a frozen interface for evaluating and improving WMs and WAMs for embodied tasks. Section 3.1 presents AoE's overview and our protocol for evaluating the executability of future videos generated for unseen tasks. Section 3.2 introduces the AoE-based vision-to-action (VITA) loss and its integration into world-model and WAM training. Section 3.3 analyzes how this loss guides future prediction and improves downstream execution.

3.1 CAMERA-SPACE INVERSE DYNAMICS FOR CROSS-TASK EVALUATION

An IDM infers actions that connect a current observation o_t to a future observation o_{t+i}. Predicting pose-based actions directly in environment coordinates ties this mapping to the coordinate convention of the training environment: the same end-effector motion can correspond to different action values under different world or robot-base frames. The action head consequently learns both the motion implied by the images and an environment-specific coordinate mapping.

We separate these two operations with a camera-space action representation. For arm $r \in \{\mathcal{L}, \mathcal{R}\}$, AoE predicts

$$\hat{a}^r_{t,t+i} = \mathcal{D}^r_\phi(o_t, o_{t+i}; s_t) \in \mathbb{R}^{d_a}, \tag{5}$$

where s_t is the current robot state, and d_a denotes the dimension of the camera-space action target. AoE summarizes each arm's motion $\hat{a}^r_{t,t+i}$ by its projected start and end positions, relative depth change, orientation change, and gripper command from timestamp t to $t+i$. This compact target captures the transition between the two observations and provides a common prediction space across environments. The target encoding and its conversion into executable poses using camera calibration are specified in Appendix B; the complete architecture and token routing are given in Appendix C.

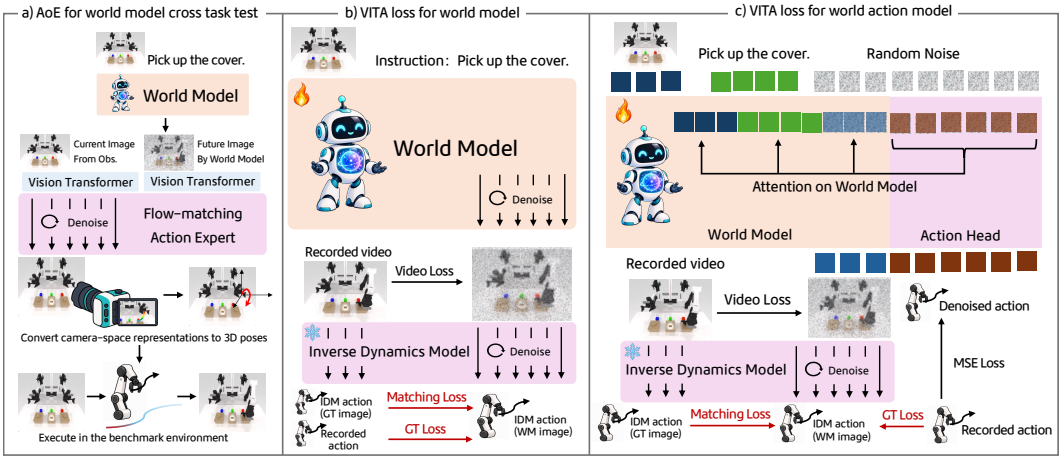


Figure 2: Cross-task generalization evaluation and the VITA loss. *Left:* Future videos generated by the WM in unseen environments are fed into AoE frame-by-frame to obtain camera-space representations, which are translated into 3D poses and executed in the real environment for evaluation. *Middle:* Dreamed future frames are decoded by AoE into actions, and the WM is updated by minimizing their discrepancies against actions inferred from recorded videos as well as the ground-truth actions. *Right:* The vision-to-action loss is transferred to WAM. By encouraging the future video to focus on task-critical regions, the native action generation of WAM is improved under cross-task generalization.

Through large-scale pretraining on robot interaction data, AoE learns to recover end-effector motion from paired observations in this shared representation. Environment-specific coordinate conversion is handled explicitly at execution, allowing the learned image-to-action mapping to be reused across downstream environments. AoE remains frozen after pretraining.

Evaluating cross-task future videos. Given an initial observation and a task instruction, a WM generates a sequence of future observations. We apply AoE to consecutive frame pairs, recover the corresponding pose commands, and execute them to measure task success. This turns predicted videos into an execution-based test of cross-task generalization.

We first evaluate the same procedure on recorded videos from tasks excluded from AoE pretraining, establishing the action-recovery model's transfer performance. We then substitute world-model predictions while keeping AoE, the initial conditions, and the execution protocol fixed. If the generated futures preserve the information needed for control and AoE reliably recovers the corresponding actions, this pipeline supports zero-shot execution on unseen tasks. The comparison with recorded videos measures how replacing genuine visual transitions with predicted ones affects downstream success.

3.2 Vision-to-Action Future Supervision

World-model video objectives treat visual content broadly, while downstream manipulation depends strongly on the relative geometry of the gripper and the target object. AoE provides an action-grounded training signal for this geometry. We use the pretrained AoE model as a frozen supervisor during world-model training, so that the generated future is evaluated according to the action information it conveys.

Given the current observation o_t, let o_{t+i} denote the recorded future observation and let $\hat{o}_{t+i}$ denote the corresponding world-model prediction. We pass both observation pairs through AoE under the same robot state, noisy action, and flow time. Let $f_\phi(o_t, o)$ denote AoE's action-flow prediction conditioned on the observation pair (o_t, o), and let $\mathbf{v}^\star_{t,i}$ denote the ground-truth action-flow target. The core objectives are

$$\mathcal{L}_{\text{match}} = \mathbb{E}\left[\left\|f_\phi(o_t, \hat{o}_{t+i}) - f_\phi(o_t, o_{t+i})\right\|_2^2\right], \quad \mathcal{L}_{\text{GT}} = \mathbb{E}\left[\left\|f_\phi(o_t, \hat{o}_{t+i}) - \mathbf{v}^\star_{t,i}\right\|_2^2\right], \tag{6}$$

where the expectation averages over training transitions and the shared action-noising conditions. We refer to their sum, $\mathcal{L}_{\text{VITA}} = \mathcal{L}_{\text{match}} + \mathcal{L}_{\text{GT}}$, as the VITA loss.

The matching term transfers action-relevant information from the recorded future, while the ground-truth (GT) term anchors this information to the action that produced the transition. We add $\mathcal{L}_{\text{VITA}}$ to the native world-model objective during cross-task training. Gradients are propagated through the generated-image branch while AoE remains frozen, allowing the WM to improve the visual details that affect action recovery. The flow-target construction and implementation details are provided in Appendices D and I.

3.3 How vision-to-action Supervision Shapes Prediction and Control

We analyze how the VITA loss changes future prediction and how these changes reach the native action head. The expressions below condition on one transition and average over the shared AoE noising conditions introduced in Section 3.2. Detailed conditions and proofs are provided in Appendix E.

Supervising action-relevant visual details. The matching and GT losses jointly guide AoE's prediction toward the average of the recorded-future prediction and the ground-truth action-flow target:

$$\bar{\mathbf{v}} = \frac{f_\phi(o_t, o_{t+i}) + \mathbf{v}^\star_{t,i}}{2}, \quad \mathcal{L}_{\text{VITA}} = \mathcal{L}_{\text{match}} + \mathcal{L}_{\text{GT}} = 2\mathbb{E}\left[\|f_\phi(o_t, \hat{o}_{t+i}) - \bar{\mathbf{v}}\|_2^2\right] + C, \quad (7)$$

where C is constant with respect to the WM. Let $J_\phi(o) = \partial f_\phi(o_t, o)/\partial o$. For a small visual perturbation Δo,

$$\mathcal{L}_{\text{match}}(o_{t+i} + \Delta o) = \Delta o^\top \mathbb{E}\left[J_\phi(o_{t+i})^\top J_\phi(o_{t+i})\right] \Delta o + O(\|\Delta o\|_2^3). \quad (8)$$

Thus visual errors receive stronger penalties when they have a larger effect on action recovery. Appendix A visualizes AoE's attention to end-effector regions.

From future images to native actions. Let h denote the shared conditioning features used for future-observation and native-action prediction, and let θ denote the parameters that produce them. Applying the chain rule to Equation (7) propagates the action-flow discrepancy through AoE and the predicted observation into h, then into θ. This updates the conditioning available to the native action head while AoE remains frozen:

$$\nabla_h \mathcal{L}_{\text{VITA}} = 4\mathbb{E}\left[\left(\frac{\partial \hat{o}_{t+i}}{\partial h}\right)^\top J_\phi(\hat{o}_{t+i})^\top \left(f_\phi(o_t, \hat{o}_{t+i}) - \bar{\mathbf{v}}\right)\right], \nabla_\theta \mathcal{L}_{\text{VITA}} = \left(\frac{\partial h}{\partial \theta}\right)^\top \nabla_h \mathcal{L}_{\text{VITA}}. \quad (9)$$

The action head continues to learn through the native objective on these updated features.

A local action-error bound. Suppose reference features $h^\star$ reproduce the recorded future and AoE captures the feature directions used by the action head. Under the local smoothness conditions in Appendix E,

$$\|\hat{\mathbf{a}} - \mathbf{a}^\star\|_2 \le \kappa \left(\underbrace{\sqrt{\frac{\mathcal{L}_{\text{VITA}}}{2}}}_{\text{VITA fitting term}} + \underbrace{\frac{\epsilon_{\text{AoE}}}{2}}_{\text{teacher error term}} \right) + \underbrace{C_h \|h - h^\star\|_2^2}_{\text{local nonlinear remainder}} + \underbrace{\epsilon_{\text{head}}}_{\text{reference head error}}, \quad (10)$$

where $\hat{\mathbf{a}}$ is the native action prediction, $\mathbf{a}^\star$ is its target, ϵ_{head} is the head's reference error, ϵ_{AoE} is the teacher's action-flow error on the recorded transition, κ relates their feature sensitivities, and C_h bounds local curvature. The bound connects accurate AoE pretraining, auxiliary fitting of generated futures, and native action learning through their respective error terms.

4 Experiments

We conduct extensive experiments on both the decoupled WM + IDM architecture and WAMs, aiming to answer the following questions:

Q1: Do world-model-generated future videos support execution on unseen tasks? (Section 4.1).

Table 1: Cross-task execution on LIBERO and RoboTwin 2.0. World models are trained on LIBERO-90 after excluding evaluation-task identities and RoboTwin with two-fold task-level cross-validation. Zero-shot rows use world-model-generated future videos, while Dataset image replay+AoE uses recorded dataset observations and Dataset action replay directly replays action trajectories from the dataset. The checkmark indicates that the frozen AoE model is used to recover actions from images.

				LIBERO (%)					RoboTwin (%)	
Model	**IDM action**	**Source**	**Inference**	**Spatial**	**Object**	**Goal**	**Long**	**Avg.**	**Easy**	**Hard**
Dataset action replay	×	Replay	–	**80.4**	**88.8**	**62.0**	**57.8**	**72.3**	**66.7**	**70.0**
Dataset image replay + AoE	✓	Replay	–	71.6	74.0	53.0	29.2	57.0	65.3	60.7
Cosmos + AoE	✓	Zero-shot	pass@1	0.4	0.6	0.4	0.0	0.4	1.6	1.1
	✓	Zero-shot	pass@50	5.2	3.8	4.0	10.6	5.9	5.3	4.7
WAN2.2 + AoE	✓	Zero-shot	pass@1	0.2	0.8	0.2	0.0	0.3	1.9	1.3
	✓	Zero-shot	pass@50	4.4	5.2	4.8	3.8	4.6	6.6	5.8

Table 2: Cross-environment execution on LIBERO-Plus. All world models are trained on LIBERO and evaluated on LIBERO-Plus. Zero-shot rows use world-model-generated future videos, while Dataset image replay+AoE uses recorded dataset observations and Dataset action replay directly replays action trajectories from the dataset. The checkmark indicates that the frozen AoE model is used to recover actions from images.

Model	**IDM action**	**Source**	**Inference**	**Background**	**Light**	**Language**	**Noise**	**Layout**	**Camera**	**Robot Init.**	**Overall**
Dataset action replay	×	Replay	–	**71.9**	**75.3**	**78.0**	**83.8**	**78.8**	**67.7**	–	**76.2**
Dataset image replay + AoE	✓	Replay	–	60.7	60.3	59.6	0.0	53.6	57.0	–	47.0
Cosmos + AoE	✓	Zero-shot	pass@1	1.1	0.4	0.5	0.0	0.1	0.4	0.3	0.4
	✓	Zero-shot	pass@50	8.7	2.8	4.2	0.2	2.7	5.4	2.1	3.5
WAN2.2 + AoE	✓	Zero-shot	pass@1	0.9	0.5	0.5	0.0	0.5	0.7	0.3	0.5
	✓	Zero-shot	pass@50	5.7	3.8	5.1	0.4	3.0	6.4	**2.5**	3.8

Q2: Can the proposed VITA loss help world models achieve cross-task generalization, and can it transfer to WAMs, improving their native action inference? (Section 4.2).

Q3: How does AoE's pretraining data scale affect its roles as evaluator and teacher? (Section 4.3).

Q4: How do AoE's action representation and the components of VITA affect performance? (Section 4.4).

Setup. We experiment with two world models, Cosmos (NVIDIA et al., 2025) and WAN2.2 (Wan et al., 2025; Wan Team, 2025), and two WAMs, LingBot-VA (Li et al., 2026a) and FastWAM (Yuan et al., 2026). For cross-task generalization, models are trained on the LIBERO-90 training split with the test tasks held out, and evaluated in the standard LIBERO (Liu et al., 2023). For RoboTwin 2.0 (Chen et al., 2025), we adopt two-fold cross-validation: 25 tasks are drawn from the clean and randomized settings for training, and the remaining 25 tasks are used for testing. For cross-environment generalization and WAM evaluation, models are trained on LIBERO and evaluated in LIBERO-Plus (Fei et al., 2026) environments. RoboTwin clean→clean+random and clean+random→clean+random are trained on clean-only and clean+random data, respectively; all other settings are kept identical. Detailed AoE, WM, and WAM training settings are provided in Appendices F and G.

4.1 How Executable Are World Model Futures on Unseen Tasks?

To evaluate the executability of world-model-generated future videos across tasks and environments, we generate future videos with Cosmos (NVIDIA et al., 2025) and WAN2.2 (Wan et al., 2025; Wan Team, 2025) from held-out initial observations and instructions, and recover actions with the frozen AoE. Table 1 shows that the evaluation itself is sound: on recorded videos, AoE recovers most of the replay performance, reaching 57.0% average success on LIBERO against 72.3% for replaying recorded actions, with RoboTwin following the same pattern. Generated futures, however, collapse. Both world models succeed on at most 0.4% of LIBERO trials under pass@1. The gap persists under cross-environment evaluation (Table 2), where recorded images still reach 47.0% overall on LIBERO-Plus while generated futures stay under 4%. Inspecting the failed generations reveals why: both models spend their capacity on task-irrelevant background, while the gripper is generated with flat appearance, physically implausible finger configurations, and hallucinated content that contradicts the observed scene, and these defects translate directly into erroneous recovered poses.

Table 3: Effect of VITA supervision on cross-task future-video executability. Zero-shot rows use future videos generated by the corresponding world model, while the replay row uses recorded dataset observations. Rows with "+VITA" use the additional action-grounded supervision. The checkmark indicates that the AoE is used to recover actions from images. Green/Red numbers show the improvement/degradation over baseline.

			LIBERO (%)					RoboTwin (%)	
Image source	**IDM action**	**Source**	**Spatial**	**Object**	**Goal**	**Long**	**Avg.**	**Easy**	**Hard**
Dataset image	✓	Replay	**71.6**	74.0	**53.0**	**29.2**	**57.0**	**65.3**	**60.7**
WAN2.2	✓	Zero-shot	0.2	0.8	0.2	0.0	0.3	1.9	1.3
Cosmos	✓	Zero-shot	0.4	0.6	0.4	0.0	0.4	1.6	1.1
WAN2.2 + VITA	✓	Zero-shot	64.4 (+64.2)	65.2 (+64.4)	36.8 (+36.6)	21.4 (+21.4)	47.0 (+46.7)	59.6 (+57.7)	40.4 (+39.1)
Cosmos + VITA	✓	Zero-shot	63.6 (+63.2)	**79.2** (+78.6)	40.6 (+40.2)	23.2 (+23.2)	51.7 (+51.3)	57.7 (+56.1)	36.4 (+35.3)

Table 4: Effect of VITA supervision under cross-environment evaluation on LIBERO-Plus. All world models are trained on LIBERO and evaluated on LIBERO-Plus. Zero-shot rows use future videos generated by the corresponding world model, while the replay row uses recorded dataset observations. Rows with "+VITA" use the additional action-grounded supervision. The checkmark indicates that the frozen AoE model is used to recover actions from images. Green/Red numbers show the improvement/degradation over baseline.

Image source	**IDM action**	**Source**	**Background**	**Light**	**Language**	**Noise**	**Layout**	**Camera**	**Robot Init.**	**Overall**
Dataset image	✓	Replay	**60.7**	**60.3**	**59.6**	0.0	**53.6**	**57.0**	–	**47.0**
WAN2.2	✓	Zero-shot	0.9	0.5	0.5	0.0	0.5	0.7	0.3	0.5
Cosmos	✓	Zero-shot	1.1	0.4	0.5	0.0	0.1	0.4	0.3	0.4
WAN2.2 + VITA	✓	Zero-shot	38.5 (+37.6)	45.3 (+44.8)	37.1 (+36.6)	**0.1** (+0.1)	39.7 (+39.2)	26.6 (+25.9)	13.4 (+13.1)	29.9 (+29.4)
Cosmos + VITA	✓	Zero-shot	40.6 (+39.5)	48.2 (+47.8)	34.1 (+33.6)	0.0 (+0.0)	33.8 (+33.7)	33.8 (+33.4)	**28.6** (+28.3)	30.3 (+29.9)

4.2 VITA Supervision Makes Futures Executable

Cross-task executability. We add the VITA loss of Section 3.2 to concentrate generation on action-relevant regions. Table 3 shows that this simple change makes generated futures executable: WAN2.2 reaches 47.0% average success on LIBERO and Cosmos 51.7%, approaching the 57.0% of recorded images, and RoboTwin Hard rises from around 1% to 40.4% and 36.4%, respectively. The gains persist across environments (Table 4), where both models reach about 30% overall on LIBERO-Plus. To test whether this improvement depends on using AoE as both the VITA teacher and the execution evaluator, we keep the same VITA-supervised WAN2.2 checkpoint and replace only the IDM used after video generation. We further evaluate VITA-trained futures with J-IDM (Li et al., 2026b) and the DreamGen IDM (Jang et al., 2025); detailed results are reported in Appendix H.

VITA for world action models. Adding $\mathcal{L}_{\text{VITA}}$ to LingBot-VA and FastWAM training improves both architectures in-distribution (Table 5); on RoboTwin, for example, LingBot-VA rises from 84.9% to 91.8% average success. The benefits are substantially larger out of distribution (Table 6): LingBot-VA improves from 35.1% to 47.2% overall on LIBERO-Plus, with the largest gains under background and lighting changes, and raises RoboTwin Hard success from 19.2% to 31.6%. FastWAM follows the same trend, which is notable because its inference path never renders future images, and the improvement arrives purely through training, as analyzed in Section 3.3.

4.3 Scaling AoE Pretraining

Both the evaluation and the supervision above rely on AoE, so we study how its pretraining data scale affects each role by partitioning the corpus into mini-sets and adding them successively, spanning 500 to 9,500 hours. As Figure 3 (left) shows, the gripper pixel error on held-out tasks drops from 12.6 px to 3.6 px while the zero-shot success rises from 1.7% to 57.0%. The converged VITA loss descends together with the pretrain loss across data scales (middle), remaining a well-behaved proxy as the teacher improves. Downstream (right), LingBot-VA's LIBERO-Plus success with a frozen AoE grows monotonically with teacher quality and overtakes the no-VITA baseline between 3,500 and 5,000 hours, whereas a trainable AoE stays nearly flat. Since updating the teacher erodes the calibrated interface that makes the supervision meaningful, AoE remains frozen in all experiments.

4.4 Ablations

We ablate three design choices: the camera-space action representation, the matching component of the VITA loss, and the cost of camera-space actions when cross-task transfer is not required.

Table 5: In-distribution performance of WAMs with VITA. Models are trained on LIBERO and RoboTwin clean+random → clean+random. Green/Red numbers show the improvement/degradation over baseline.

	LIBERO					RoboTwin: C+R → C+R		
Method	**Long**	**Goal**	**Object**	**Spatial**	**Avg.**	**Easy**	**Hard**	**Avg.**
FastWAM	95.2	97.0	**100.0**	98.2	97.6	86.4	88.2	87.3
LingBot-VA	98.5	97.2	99.6	98.5	98.5	85.6	84.2	84.9
FastWAM + VITA	97.2 (+2.0)	**98.8** (+1.8)	99.6 (-0.4)	**99.0** (+0.8)	98.7 (+1.1)	90.7 (+4.3)	**91.9** (+3.7)	91.3 (+4.0)
LingBot-VA + VITA	**98.8** (+0.3)	98.4 (+1.2)	99.8 (+0.2)	**99.0** (+0.5)	**99.0** (+0.5)	**92.4** (+6.8)	91.2 (+7.0)	**91.8** (+6.9)

Table 6: Out-of-distribution performance of WAMs with VITA. Models are trained on LIBERO and RoboTwin clean environments, then evaluated on LIBERO-Plus and RoboTwin clean+random environments, respectively. Results are success rates in percent. RoboTwin Avg. is the unweighted mean of Easy and Hard. Green/Red numbers show the improvement/degradation over the corresponding baseline in percentage points.

	LIBERO-Plus								RoboTwin: C → C+R		
Method	**Background**	**Light**	**Language**	**Noise**	**Layout**	**Robot Init.**	**Camera**	**Overall**	**Easy**	**Hard**	**Avg.**
FastWAM	53.7	78.2	68.9	**37.7**	60.7	44.5	**16.4**	51.5	69.5	9.3	39.4
LingBot-VA	56.3	46.5	61.7	11.4	34.8	36.2	9.5	35.1	74.3	19.2	46.8
FastWAM + VITA	74.6 (+20.9)	**80.5** (+2.3)	**75.6** (+6.7)	27.4 (-10.3)	**80.9** (+20.2)	**53.4** (+8.9)	14.2 (-2.2)	**55.9** (+4.4)	73.8 (+4.3)	17.5 (+8.2)	45.7 (+6.3)
LingBot-VA + VITA	**77.9** (+21.6)	71.2 (+24.7)	69.9 (+8.2)	15.0 (+3.6)	51.2 (+16.4)	49.3 (+13.1)	12.7 (+3.2)	47.2 (+12.1)	**82.5** (+8.2)	**31.6** (+12.4)	**57.1** (+10.3)

Camera-space vs. 3D-pose actions. The 3D-pose variant fits its training environments better than AoE, but this fit comes from memorizing environment-specific coordinate conventions: it recovers executable actions on unseen tasks at 0.0% zero-shot and 2.3% with 10-shot adaptation, where the camera-space representation reaches 29.2% and 37.8% (Table 7). Because the camera-space target is defined by observable image geometry, the same action interface transfers across environments without relearning a task-specific world-frame mapping.

What does camera space cost without transfer? Figure 4 compares six IDM variants as VITA supervisors within a fixed environment. Pretraining matters more than the representation: from-scratch IDMs of either type supervise poorly, while a two-stage trained 3D-pose IDM is competitive. Weak supervision is worse than none: both from-scratch variants fall on LIBERO-Plus below the baseline trained without VITA, indicating that an uncalibrated teacher actively misdirects the world model. Adapting the IDM to the downstream environment in a second stage recovers much of this deficit, yet still lags large-scale pretraining on LIBERO-Plus, suggesting that per-environment adaptation alone does not match the supervision quality of a broadly pretrained teacher. 3D pose performs better because camera-space supervision becomes unreliable under severe occlusion. The camera-space pretrained AoE remains the best supervisor on LIBERO-Plus at 47.2%, and trails 3D-pose 2-stage only on RoboTwin Hard.

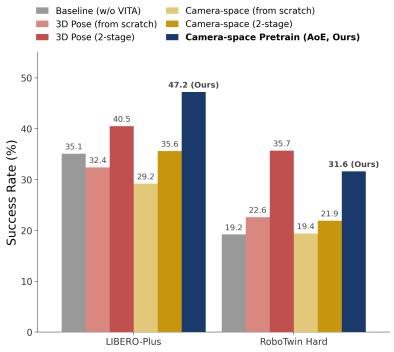


Figure 4: VITA loss computed with different IDM variants. IDM is not suitable for updating with world models. 3D pose performs better, especially in environments with severe screen occlusion.

Why the matching loss. The GT loss alone reaches 45.5% on LIBERO-Plus but only 24.7% on RoboTwin Hard (Table 8). The reason is structural: when the gripper leaves the camera view or the camera undergoes deformation, camera-space GT targets become undefined and provide little signal. The matching loss transfers action-relevant information from the recorded future without requiring a pose estimate and further disambiguates multi-solution observations, so the full VITA objective performs best on both benchmarks.

5 Related Work

Grounded world-model evaluation. A growing body of work evaluates world models in embodied scenarios. WorldArena jointly scores perceptual quality and functional utility, revealing a persistent gap between visual fidelity and embodied usefulness (Shang et al., 2026). KineBench grounds

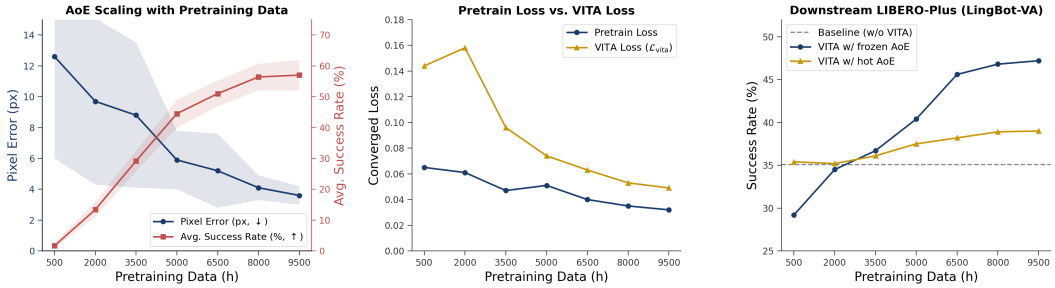


Figure 3: AoE pretraining scaling: effects on cross-task action recovery and VITA-based policy learning. *Left:* IDM's gripper pixel error and recovered-action success on cross-task dataset image replay task as pretraining data grows. *Middle:* WM's VITA loss across data scales with the converged pretrain loss. *Right:* WAM's downstream LIBERO-Plus success with the VITA loss, with AoE frozen or updated (hot); the dashed line marks WAM without VITA.

Table 7: Camera-space vs. 3D-pose action representations for inverse dynamics. Pretrain loss and validation RMSE measure action-prediction quality (↓), while cross-task success rates measure downstream execution (↑).

	Pretraining metrics		Cross-task SR (%) ↑	
Action repr.	**Pretrain loss ↓**	**Valid RMSE ↓**	**Zero-shot**	**10-shot**
Camera-space (AoE)	0.047	**0.070**	**29.2**	**37.8**
3D pose	**0.026**	0.146	0.0	2.3

Table 8: Ablation of the VITA loss components on LingBot-VA. Success rates on LIBERO-Plus and the hard split of RoboTwin when training with each aspect of VITA loss.

Model	**LIBERO-Plus (%)**	**RoboTwin Hard (%)**
LingBot-VA	35.1	19.2
+ matching loss only	36.7	23.9
+ GT loss only	<u>45.5</u>	<u>24.7</u>
+ VITA loss (Ours)	**47.2**	**31.6**

generated videos kinematically, extracting end-effector poses from individual frames for closed-loop execution and trajectory-level analysis (Liu et al., 2026b), and WorldSimProbe issues controlled simulator queries to test action realization and interaction grounding (Co et al., 2026). Despite this progress, existing evaluations remain confined to imagination: they score the generated videos themselves, or execute them only through extractors tied to their training environments, leaving unquantified whether a predicted future can be carried into an unseen environment and actually executed (Jiang et al., 2026). We therefore use dataset-image and recorded-action replay to calibrate IDM transfer, and hold the pretrained IDM fixed across visual-source replacements, human judgments, and local geometric interventions.

Video and inverse-dynamics priors for control. DreamGen converts generated videos into neural trajectories for policy training (Jang et al., 2025). LingBot-VA couples causal video modeling with action prediction (Li et al., 2026a), while Fast-WAM studies video co-training under several inference interfaces (Yuan et al., 2026). VERA combines video planning with a Jacobian-based inverse dynamics model (Li et al., 2026b). GE-Act 2.0 separately pretrains a visual planner and IDM, then jointly optimizes them through generated future states using action flow matching and compatibility-based selection (AgiBot Research Team et al., 2026). These approaches establish visual prediction and inverse dynamics as complementary components of robot learning. We evaluate a pretrained, frozen camera-space IDM through dataset-image replay on held-out tasks and transfer its action knowledge into future-image supervision.

6 Conclusion

Pretrained inverse dynamics provides a common action model for evaluating and supervising visual futures. We introduce AoE, an inverse-dynamics model built on a camera-projection action representation with zero-shot cross-task generalization, through which world-model futures are carried from video into interactive execution in the environment. Building on this interface, we further propose the VITA loss, which substantially improves the cross-task generalization of world models. We also extend VITA to world action models, obtaining consistent gains across multiple architectures and evaluation environments. Constrained by the camera-projection representation, AoE cannot handle tasks where the gripper leaves the camera view for extended periods or the camera is severely occluded, and would raise additional safety concerns in real-robot evaluation; we expect these limitations to be addressed in combination with future work on active exploration.

AI USE STATEMENT

In this work, we used generative AI tools to assist with checking the training and evaluation data pipelines and code, polishing the language of the manuscript, and developing and checking the mathematical derivations for the VITA loss. We have not used generative AI tools for any other tasks requiring disclosure, and the remaining required disclosure categories are not applicable to this work. All AI-assisted work was reviewed and confirmed by the main authors against the underlying code, data, experiments, and mathematical derivations. We take full responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed training and evaluation configurations in Appendix F, including the composition and proportions of the pretraining data in Appendix F.1, and detailed experimental results in Appendix G. The equations for converting camera-space predictions into 3D poses in the robot coordinate frame are provided in Appendix B. A formal analysis of the VITA loss, including its assumptions and proofs, is presented in Appendix E. The supplementary materials include all training and evaluation code, together with the code for converting and cleaning the pretraining data. We also plan to publicly release the pretrained AoE checkpoint in the future.

REFERENCES

AgiBot Research Team, Renhang Liu, Wenzhi Zhao, Zhuo Yang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Guanghui Ren, Youlun Peng, Rongjun Jin, Nan Wang, Sukai Wang, Xindong He, Jinyuan Feng, Ziyu Xiong, Linqing Zhong, Yifei Wei, Feng Han, Long Zhang, Da Huang, Nanshu Zhao, Chenghao Yin, Mo Wu, Zhaodong Yan, Kongtao Hu, Yuxiang Yan, Aogelijiang Niyazi, Yu Fang, Jia Zeng, Lizhu Meng, Daizhen Lv, Haoyu Cao, Zhiwen Hou, Lianjin Ye, Yuehan Niu, Zhikai Cai, Xuan Hu, Hui Min, Xiongfeng Cai, Yue Liao, Jing Wu, Soujanya Poria, Ye Li, Sanping Zhou, and Maoqing Yao. GE-Act 2.0: Pretraining and scaling a world-action model for robotic manipulation. *arXiv preprint arXiv:2609.05588*, 2026. URL https://arxiv.org/abs/2609.05588.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025. URL https://arxiv.org/abs/2506.18088.

Peterson Co, Sicheng Hu, Chunxuan Jiao, Hongyang Cheng, Yulin Luo, Yijie Xu, Sixiang Chen, Zhongxia Zhao, Zihao Wang, DaFeng Chi, Peidong Liu, YuTong Chen, Henghua Liu, Zhihao Yuan, Huizhu Jia, Yuzheng Zhuang, Tianle Zhang, Liang Lin, Huajie Tan, and Shanghang Zhang. WorldSimProbe: Diagnosing simulator faithfulness in action-conditioned world models for embodied manipulation. *arXiv preprint arXiv:2608.09298*, 2026. URL https://arxiv.org/abs/2608.09298.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38574–38583, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Fei_LIBERO-Plus_A_Progressive_Robustness_Benchmark_for_Visual-Language-Action_Models_CVPR_2026_paper.html.

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, 2025. URL https://arxiv.org/abs/2512.17661.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loïc Magne, Ajay Mandlekar, Avnish Narayan,

You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. DreamGen: Unlocking generalization in robot learning through video world models. In *Proceedings of the 9th Conference on Robot Learning*, pp. 5170–5194. PMLR, 2025. URL https://proceedings.mlr.press/v305/jang25a.html.

Feng Jiang, Yang Chen, Kyle Xu, Yuchen Liu, Haifeng Wang, Zhenhao Shen, Jasper Lu, Shengze Huang, Yuanfei Wang, Chen Xie, and Ruihai Wu. RoboWM-Bench: A benchmark for evaluating world models in robotic manipulation, 2026. URL https://arxiv.org/abs/2604.19092.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026a. URL https://arxiv.org/abs/2601.21998.

Sizhe Lester Li, Evan Kim, Xingjian Bai, Tong Zhao, Tao Pang, Max Simchowitz, and Vincent Sitzmann. Turning video models into generalist robot policies. *arXiv preprint arXiv:2605.27817*, 2026b. URL https://arxiv.org/abs/2605.27817.

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

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2306.03310.

Zeyi Liu, Shuang Li, Eric Cousineau, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Geometry-aware 4D video generation for robot manipulation, 2026a. URL https://arxiv.org/abs/2507.01099.

Zeyu Liu, Zhangzhe Zhu, Yang Zhang, Chenyou Fan, Chenjia Bai, and Xuelong Li. KineBench: Benchmarking embodied world models via IDM-free kinematic grounding. *arXiv preprint arXiv:2607.19876*, 2026b. URL https://arxiv.org/abs/2607.19876.

Weishi Mi, Yong Bao, Xiaowei Chi, Xiaozhu Ju, Zhiyuan Qin, Kuangzhi Ge, Kai Tang, Peidong Jia, Shanghang Zhang, and Jian Tang. TC-IDM: Grounding video generation for executable zero-shot robot motion, 2026. URL https://arxiv.org/abs/2601.18323.

NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025. URL https://arxiv.org/abs/2501.03575.

Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, Xin Jin, Ziyou Wang, Lei Jin, Xin Zhang, Yinzhou Tang, Haisheng Su, Chen Gao, Wei Wu, Xihui Liu, Dhruv Shah, Zhaoxiang Zhang, Zhibo Chen, Jun Zhu, Yonghong Tian, Tat-Seng Chua, Wenwu Zhu, and Yong Li. WorldArena: A unified benchmark for evaluating perception and functional utility of embodied world models. *arXiv preprint arXiv:2602.08971*, 2026. URL https://arxiv.org/abs/2602.08971.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel

Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. URL https://arxiv.org/abs/2508.10104.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. URL https://arxiv.org/abs/1812.01717.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Wan Team. Wan2.2: Official code and model release. Official GitHub repository, 2025. URL https://github.com/Wan-Video/Wan2.2.

Ruixiang Wang, Qingming Liu, Yueci Deng, Guiliang Liu, Zhen Liu, and Kui Jia. Eva: Aligning video world models with executable robot actions via inverse dynamics rewards, 2026. URL https://arxiv.org/abs/2603.17808.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13 (4):600–612, 2004. doi: 10.1109/TIP.2003.819861. URL https://ece.uwaterloo.ca/~z70wang/publications/ssim.html.

Youngsun Wi, Mark Van der Merwe, Pete Florence, Andy Zeng, and Nima Fazeli. CALAMARI: Contact-aware and language conditioned spatial action mapping for contact-rich manipulation. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2753–2771. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/wi23a.html.

Jimmy Wu, Xingyuan Sun, Andy Zeng, Shuran Song, Johnny Lee, Szymon Rusinkiewicz, and Thomas Funkhouser. Spatial action maps for mobile manipulation, 2020. URL https://arxiv.org/abs/2004.09141.

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. URL https://arxiv.org/abs/2603.16666.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. URL https://arxiv.org/abs/1801.03924.

Haoyu Zhen, Zixian Gao, Qiao Sun, Yilin Zhao, Yuncong Yang, Yilun Du, Pengsheng Guo, Tsun-Hsuan Wang, Yi-Ling Qiao, and Chuang Gan. Action images: End-to-end policy learning via multiview video generation, 2026. URL https://arxiv.org/abs/2604.06168.

Yupeng Zheng, Xiang Li, Songen Gu, Yuhang Zheng, Shuai Tian, Weize Li, Linbo Wang, Chaoyue Li, Qichao Zhang, Haoran Li, Zhongpu Xia, Ya-Qin Zhang, Shuicheng Yan, and Dongbin Zhao. GIFT: Guided intermediate feature training via action-oriented structural supervision for robotic manipulation, 2026. URL https://arxiv.org/abs/2609.04193.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer*

Vision and Pattern Recognition, pp. 5745–5753, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Zhou_On_the_Continuity_of_Rotation_Representations_in_Neural_Networks_CVPR_2019_paper.html.

A Attention Maps and Camera-Space Representations on Seen and Unseen Tasks

We visualize AoE on two types of data: a validation split sampled from the pretraining dataset and tasks from unseen environments. The same frozen AoE checkpoint is used in both cases.

Attention maps. Figure 5 shows attention maps for the validation split and unseen tasks. Across both settings, the attention is concentrated around the robot end effector and the object being manipulated, while receiving less emphasis from background regions. This behavior is consistent with the action-grounded sensitivity analyzed in Section 3.3: visual changes near the end effector and the target object have a direct effect on the action-flow prediction. The attention maps provide a qualitative visualization of this behavior, while the formal analysis in Section 3.3 is based on the local sensitivity of AoE's action-flow output to its visual input.

Camera-space representations. Figure 6 visualizes the camera-space action representation predicted by AoE. The examples are obtained from recorded videos in the validation split and unseen environments. For each observation pair, AoE predicts the two-dimensional end-effector trajectory in the camera image, shown as the overlaid line between the recorded start and end frames. The plot below each example shows the corresponding predicted relative-depth change. These visualizations show that AoE produces a consistent camera-space representation across seen and unseen tasks, without requiring the action target to be expressed in a task-specific world coordinate frame.

B Camera-Space Action Representation and Pose Reconstruction

This appendix specifies the camera-space action targets used to pretrain AoE and their conversion into executable pose commands. We use subscripts s and e for the start and end of an observation pair. Superscripts c and w denote the camera frame and the world or robot-base frame used by the controller, respectively.

Endpoint representation. For each arm $r \in \mathcal{R}$, let $\mathbf{p}_s^r$ and $\mathbf{p}_e^r$ denote the start and end positions of the end effector. We summarize the translational component of the transition by the segment

$$\mathbf{p}^r(\alpha) = (1-\alpha)\mathbf{p}_s^r + \alpha\mathbf{p}_e^r, \qquad \alpha \in [0,1]. \tag{11}$$

The action target also retains the relative orientation change and the endpoint gripper command. The segment describes the geometric displacement represented by the target; intermediate waypoints and their timing are omitted from this representation.

Projection into the camera frame. Let K denote the camera intrinsic matrix, and let $(R_{cw}, \mathbf{t}_{cw})$ transform positions from the controller frame into the camera frame. For endpoint $j \in \{s, e\}$,

$$\mathbf{p}_j^{c,r} = R_{cw}\mathbf{p}_j^{w,r} + \mathbf{t}_{cw}, \qquad \begin{bmatrix} u_j^r \\ v_j^r \\ 1 \end{bmatrix} = \frac{1}{z_j^r} K \mathbf{p}_j^{c,r}, \tag{12}$$

where z_j^r is the third coordinate of $\mathbf{p}_j^{c,r}$. The intrinsic matrix is expressed in the same pixel coordinate system as the input images.

For an image of width W and height H, we normalize the projected coordinates as

$$\bar{u}_j^r = \frac{2u_j^r}{W} - 1, \qquad \bar{v}_j^r = \frac{2v_j^r}{H} - 1. \tag{13}$$

The depth target records the change along the camera's depth axis:

$$\Delta z^r = z_e^r - z_s^r, \qquad \widetilde{\Delta z}^{\,r} = \mathcal{N}_z(\Delta z^r), \tag{14}$$

where $\mathcal{N}_z$ denotes the depth normalization used during pretraining. The same normalization and its inverse are used at evaluation.

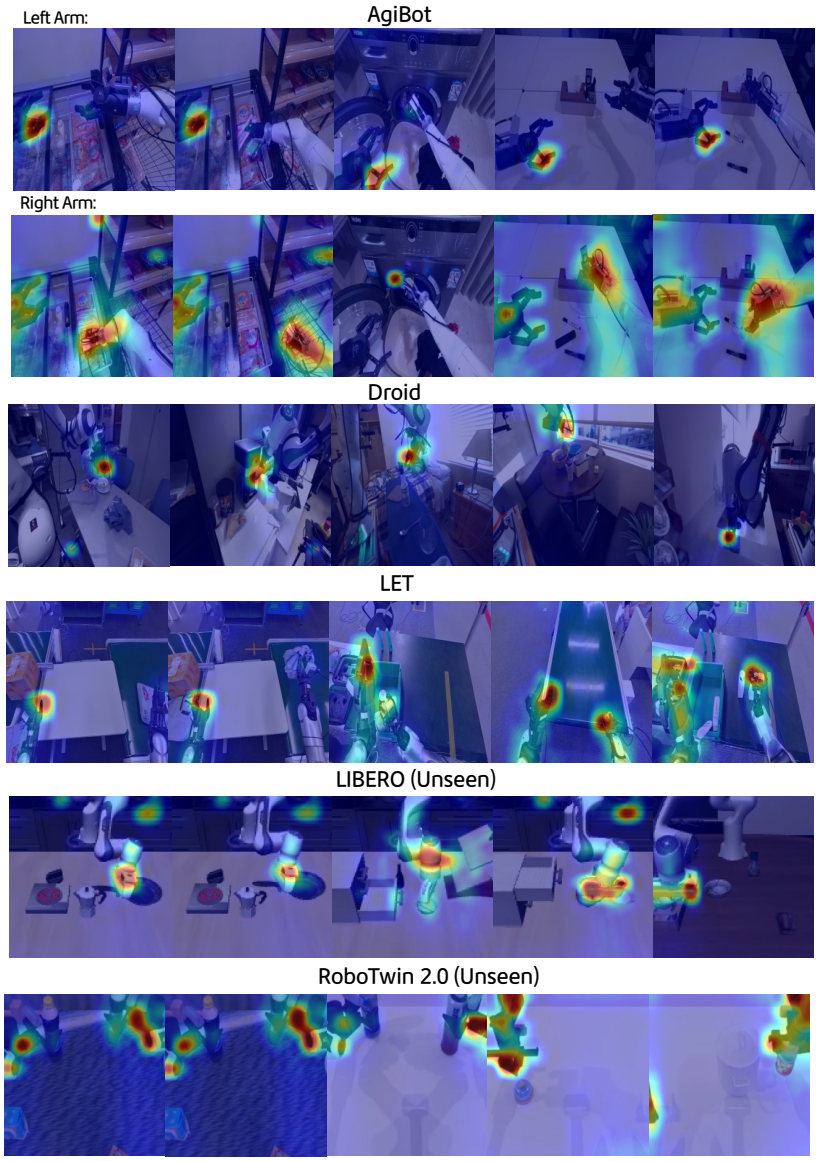


Figure 5: Attention maps on seen and unseen tasks. AoE focuses primarily on the robot end effector and the object being manipulated in both the validation split and unseen environments.

Rotation and gripper targets. Let $\mathbf{R}_s^{c,r}$ and $\mathbf{R}_e^{c,r}$ denote the end-effector orientations in the camera frame. We write the relative-rotation target as

$$\boldsymbol{\rho}^r = \mathrm{Enc}_{\mathrm{rot}}\left(\mathrm{Rel}(\mathbf{R}_s^{c,r}, \mathbf{R}_e^{c,r})\right). \tag{15}$$

The 6D rotation encoding follows Zhou et al. (2019). Here, Rel follows the relative-rotation convention used to construct the training labels, and $\mathrm{Enc}_{\mathrm{rot}}$ maps the resulting rotation to its continuous representation. We use a six-dimensional rotation encoding. Its decoder, $\mathrm{Dec}_{\mathrm{rot}}$, is paired with the same rotation convention during pose reconstruction. The scalar g^r denotes the endpoint gripper command.

The complete target for one arm is

$$\tilde{\mathbf{a}}^r = \left[\bar{u}_s^r, \bar{v}_s^r, \bar{u}_e^r, \bar{v}_e^r, \widetilde{\Delta z}^{\,r}, \boldsymbol{\rho}^r, g^r\right] \in \mathbb{R}^{d_a}. \tag{16}$$

In our implementation, $d_a = 12$: four projected endpoint coordinates, one depth-change value, six rotation coordinates, and one gripper value. Each arm has its own action target.

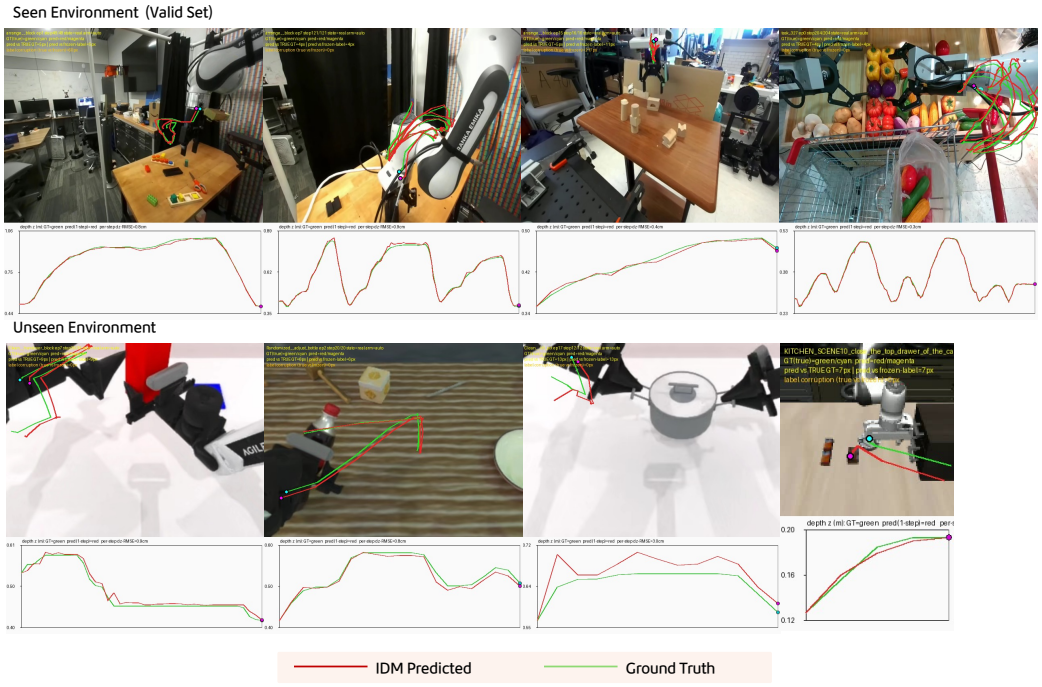


Figure 6: Camera-space representation on seen and unseen tasks. The overlaid trajectories are generated by AoE from recorded video pairs, and the plots below show the corresponding predicted relative-depth changes.

Reconstructing the endpoint position. The current robot state s_t supplies the initial end-effector position and orientation through forward kinematics. The initial camera-frame depth is

$$z_s^r = \left[R_{cw}\mathbf{p}_s^{w,r} + \mathbf{t}_{cw}\right]_z . \tag{17}$$

Given the predicted normalized endpoint coordinates and depth change, we recover

$$\hat{u}_e^r = \frac{W}{2}(\hat{\bar{u}}_e^r + 1), \qquad \hat{v}_e^r = \frac{H}{2}(\hat{\bar{v}}_e^r + 1), \qquad \hat{z}_e^r = z_s^r + \mathcal{N}_z^{-1}\left(\widehat{\widehat{\Delta z}}^{\,r}\right). \tag{18}$$

Back-projection gives the endpoint in the camera frame, followed by conversion into the controller frame:

$$\hat{\mathbf{p}}_e^{c,r} = \hat{z}_e^r K^{-1}\begin{bmatrix}\hat{u}_e^r \\ \hat{v}_e^r \\ 1\end{bmatrix}, \qquad \hat{\mathbf{p}}_e^{w,r} = R_{cw}^{\top}\left(\hat{\mathbf{p}}_e^{c,r} - \mathbf{t}_{cw}\right). \tag{19}$$

Reconstructing the endpoint orientation. The initial orientation and predicted relative rotation determine the endpoint orientation:

$$\hat{\mathbf{R}}_e^{c,r} = \mathrm{Apply}\left(\mathbf{R}_s^{c,r}, \mathrm{Dec}_{\mathrm{rot}}(\hat{\boldsymbol{\rho}}^r)\right), \qquad \hat{\mathbf{R}}_e^{w,r} = R_{cw}^{\top}\hat{\mathbf{R}}_e^{c,r}. \tag{20}$$

The operators Rel and Apply use matching conventions, satisfying $\mathrm{Apply}(R_s, \mathrm{Rel}(R_s, R_e)) = R_e$. The reconstructed position, orientation, and gripper command are then passed to the environment's controller using its action convention.

Execution from video. For a generated video, AoE processes consecutive frame pairs. At each execution step, the current robot state supplies the initial pose required for reconstruction. Camera calibration, action normalization, and controller conventions remain fixed when comparing recorded and generated videos. All coordinate conversions are deterministic, and AoE retains its pretrained parameters throughout evaluation.

C Inverse-Dynamics Architecture and Protocol Details

Visual and action streams. AoE receives two external-camera images, `ext_start` and `ext_end`. Each timestamp has an independent DINOv3 (Siméoni et al., 2025) ViT-L/16 encoder;

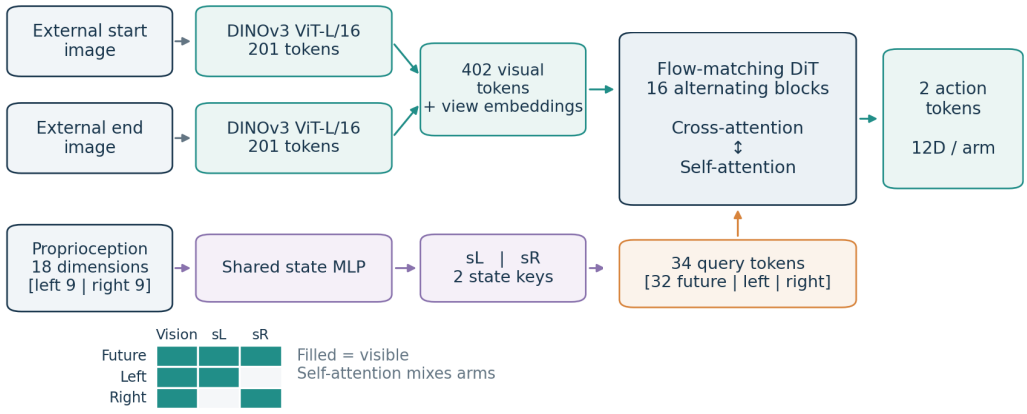


Figure 7: The AoE inverse-dynamics architecture. Two independent image towers encode external-camera observations at the start and end of a transition. Alternating attention blocks connect shared visual features to two arm-action tokens. Direct state access is routed by arm; self-attention supports coordination. The output parameterizes translation in image coordinates and depth, with rotation and gripper control.

both full encoder parameter sets are updated during AoE pretraining. Each encoder produces 201 tokens: one class token, four register tokens, and 196 patch tokens on a 14×14 grid. Learned view embeddings distinguish the timestamps, producing 402 visual conditioning tokens. The model uses no wrist-camera input (Figure 7).

The 18-dimensional proprioceptive vector is split into two nine-dimensional arm states. A shared state MLP produces tokens s_L and s_R, bringing the conditioning stream to 404 tokens. The query stream has 34 tokens: 32 future-representation tokens and two 12-dimensional arm-action tokens projected into the transformer width. The action-token axis indexes arms. The transformer has width 1536, 16 layers, 32 attention heads of dimension 48, and MLP hidden width 1024, with approximately 450 million parameters. Together with both visual encoders, AoE contains approximately 1.14 billion parameters.

State routing and positional information. Even-indexed layers use cross-attention over the visual and state conditioning stream. The left action token attends to all visual tokens and s_L; the right action token attends to all visual tokens and s_R. The 32 future tokens attend to the complete conditioning stream. Odd-indexed layers apply unmasked self-attention across queries, allowing information exchange and bimanual coordination. Thus, the cross-attention mask routes direct state access, while self-attention permits indirect cross-arm influence.

Cross-attention uses two-dimensional axial rotary positional embeddings. Image patches carry their row and column coordinates, and each query token has a learned two-dimensional reference point. Both state tokens use coordinate $(0, 0)$. The arm-supervision mask expands across all 12 action coordinates. If a minibatch contains multiple temporal pairs k, the numerator of the masked pretraining loss sums over (b, k, r) and coordinates, and its denominator is $12 \max(1, \sum_{b,k,r} m_{bkr})$. The same reduction applies at every sampled flow time. Unsupervised arms receive no direct loss contribution.

C.1 Task splits and frozen-model reuse

LIBERO source tasks are selected from LIBERO-90 after excluding evaluation-task identities. The exclusion is applied before AoE pretraining. Dataset image replay evaluates the pretrained checkpoint on recorded transitions from these unseen tasks. The world-model source comparison reuses the same checkpoint for recorded, cross-task generated, and VITA-supervised generated futures. The VITA condition updates the world model alone.

RoboTwin uses two-fold task-level cross-validation with 25 training tasks and 25 held-out tasks per fold. The two groups exchange roles between folds. Each fold reuses its pretrained AoE checkpoint with frozen parameters. The complete task partition is given in Appendix F.4.

The native WAM transfer comparison trains on LIBERO-90 and evaluates the LIBERO Long, Goal, Object, and Spatial suites, with 500 episodes per suite and 2,000 episodes in total. The All training objectives in Table 20 share this split, including the native and DINO controls. The same-domain world-model comparison uses 500 episodes, with target adaptation supplied 50 demonstrations per task. Standard LIBERO and LIBERO-Plus native-control experiments use training on benchmark tasks and are reported as a separate evaluation setting.

AoE parameters remain frozen after pretraining, except in the explicitly identified trainable-teacher scaling ablation in Section 4.3. Source-validation data provide model selection and the offline confidence weights used by the auxiliary loss. Confidence calibration changes loss weights while preserving the pretrained model. In downstream WAM training, differentiation through AoE's generated-image input transfers supervision to video and shared parameters. Each WAM uses its native action path for evaluation. The fold teacher and its confidence calibration are shared across downstream architecture comparisons.

C.2 Calibration protocol

Held-out task calibration fixes task identities before evaluation and excludes their demonstrations from AoE pretraining. Recorded actions and future images share the simulator snapshot and temporal interval. Camera perturbations are applied when rendering both images of a pair. We report full-set performance, the coverage and performance of instances with successful recorded-action replay, and their subset with successful demonstration-image recovery. An independent geometric recovery pipeline repeats the source comparison and the paired visual interventions. Rotation targets encode the start-to-end relative rotation in a consistent reference frame, and depth normalization is inverted before Eq. (19). Appendix I specifies the complete auxiliary training objective, confidence calibration, and gradient schedule.

D Implementation of the vision-to-action Loss

Action-flow supervision. AoE is pretrained with flow matching (Lipman et al., 2023). Its supervised output is an action-update vector: the direction in which a noisy action estimate should move toward the target action. For a training pair (o_t, o_{t+i}), let $\tilde{\mathbf{a}}_{t,t+i}$ denote the ground-truth camera-space action defined in Appendix B. We sample Gaussian noise $\boldsymbol{\epsilon}$ and a flow time τ, and construct

$$\mathbf{z}_\tau = (1-\tau)\boldsymbol{\epsilon} + \tau\tilde{\mathbf{a}}_{t,t+i}, \qquad \mathbf{v}^\star_{t,i} = \tilde{\mathbf{a}}_{t,t+i} - \boldsymbol{\epsilon}, \tag{21}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ and $\tau \sim \mathcal{U}[0,1]$. The vector $\mathbf{z}_\tau$ is the noisy action input, and $\mathbf{v}^\star_{t,i}$ is the corresponding target update.

The notation $f_\phi(o_t, o_{t+i})$ in the main text abbreviates AoE's flow prediction conditioned on the image pair, current robot state s_t, noisy action $\mathbf{z}_\tau$, and flow time τ. The generated-image and recorded-image branches share all conditions except the future image. Their difference therefore measures how replacing the future image changes AoE's action prediction. The recorded-image branch is evaluated without gradient tracking.

Masked and confidence-weighted errors. The implementation uses weighted robust errors for the two objectives. For prediction $\mathbf{x}$, target $\mathbf{y}$, and nonnegative coordinate weights $\mathbf{w}$, we compute

$$\mathcal{R}(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \frac{\sum_j w_j \left(\sqrt{1 + (x_j - y_j)^2} - 1\right)}{\sum_j w_j}, \tag{22}$$

over samples with nonzero total weight. Samples without valid supervised coordinates are omitted.

The matching weights combine supervision validity, coordinate selection, and offline confidence calibrated on recorded transitions. The GT weights additionally account for the teacher's reliability on the corresponding recorded image pair. Coordinates on which the teacher has a larger error against the ground-truth flow target receive less weight in the GT objective. These weights are held fixed when differentiating with respect to generated images. The projected start coordinates remain part of AoE's action representation and receive zero direct auxiliary weight.

Using these weights, the implemented action-flow objectives are

$$\begin{aligned}\mathcal{L}^{\mathrm{rob}}_{\mathrm{match}} &= \mathbb{E}\left[\mathcal{R}\left(f_\phi(o_t, \hat{o}_{t+i}), f_\phi(o_t, o_{t+i}); \mathbf{w}_{\mathrm{match}}\right)\right], \\ \mathcal{L}^{\mathrm{rob}}_{\mathrm{GT}} &= \mathbb{E}\left[\mathcal{R}\left(f_\phi(o_t, \hat{o}_{t+i}), \mathbf{v}^\star_{t,i}; \mathbf{w}_{\mathrm{GT}}\right)\right].\end{aligned} \tag{23}$$

The GT objective can produce noisy image gradients when the frozen teacher is unreliable on generated observations. To provide a direct image-space anchor, we supplement the matching objective with a weak reconstruction term around the ground-truth gripper locations. This ROI term uses Gaussian spatial weights and a Charbonnier reconstruction error. It is included as an implementation-level stabilizer alongside the matching loss.

Gradient propagation and training. Generated future frames are decoded differentiably before being passed to AoE. AoE and the video decoder retain frozen parameters, while gradients through their inputs update the world model. The two auxiliary objectives are averaged over the supervised future frames and added to the native training objective. For WAMs, the native action objective is retained, and the auxiliary gradients reach the video branch and shared parameters. Evaluation follows the original action inference path.

E Analysis of vision-to-action Supervision

Notation and scope. We first consider a single transition, holding the current observation, instruction, robot state, and world-model sampling inputs fixed. Write $y = o_{t+i}$ for the recorded future and $\hat{y}$ for the predicted future. Let ζ collect the shared AoE noising conditions, including the noisy action and flow time. We abbreviate the frozen teacher's action-flow prediction as

$$F_\zeta(y) = f_\phi(o_t, y; \zeta), \qquad q_\zeta = F_\zeta(y), \qquad b_\zeta = \frac{q_\zeta + \mathbf{v}^\star_\zeta}{2}, \tag{24}$$

where $\mathbf{v}^\star_\zeta$ is the ground-truth action-flow target. The current robot state is included in the conditioning of F_ζ. All derivatives with respect to the future image hold these conditioning variables fixed. For a quantity u_ζ, we use $\|u\|_\zeta = (\mathbb{E}_\zeta \|u_\zeta\|_2^2)^{1/2}$ to denote its root-mean-square magnitude over the shared noising conditions.

The analysis first treats the squared objectives in Section 3.2. Averaging the per-transition expressions over training transitions gives the full training objective. We discuss fixed weights and robust penalties at the end of this appendix.

Proposition 1: the combined supervision target. The sum of the matching and GT objectives satisfies

$$\begin{aligned}\mathcal{L}_{\mathrm{aux}}(\hat{y}) &= \mathbb{E}_\zeta\left[\|F_\zeta(\hat{y}) - q_\zeta\|_2^2 + \|F_\zeta(\hat{y}) - \mathbf{v}^\star_\zeta\|_2^2\right] \\ &= 2\|F(\hat{y}) - b\|_\zeta^2 + \frac{1}{2}\epsilon^2_{\mathrm{AoE}}, \qquad \epsilon_{\mathrm{AoE}} = \|q - \mathbf{v}^\star\|_\zeta.\end{aligned} \tag{25}$$

Here ϵ_{AoE} measures the frozen teacher's action-flow error on the recorded transition.

Proof. For any vectors u, q, v,

$$\|u - q\|_2^2 + \|u - v\|_2^2 = 2\left\|u - \frac{q+v}{2}\right\|_2^2 + \frac{1}{2}\|q - v\|_2^2.$$

Substituting the teacher predictions and taking expectations gives Equation (25). The final term is independent of the world-model parameters because AoE, the recorded observation, and the ground-truth action target are fixed.

This identity also gives

$$\begin{aligned}\|F(\hat{y}) - q\|_\zeta &\le \|F(\hat{y}) - b\|_\zeta + \frac{\epsilon_{\mathrm{AoE}}}{2} \\ &= \sqrt{\frac{\mathcal{L}_{\mathrm{aux}}(\hat{y})}{2} - \frac{\epsilon^2_{\mathrm{AoE}}}{4}} + \frac{\epsilon_{\mathrm{AoE}}}{2} \\ &\le \sqrt{\frac{\mathcal{L}_{\mathrm{aux}}(\hat{y})}{2}} + \frac{\epsilon_{\mathrm{AoE}}}{2}.\end{aligned} \tag{26}$$

The same bound holds for $\|F(\hat{y}) - \mathbf{v}^\star\|_\zeta$. Both follow from the triangle inequality for root-mean-square norms. Thus the generated transition's action-flow error depends on its fit to the auxiliary target and on the teacher's accuracy on the recorded transition.

Proposition 2: local weighting of visual errors. Suppose F_ζ is twice differentiable near y, with uniformly bounded first and second derivatives. Let $J_\zeta(y) = \partial F_\zeta(y)/\partial y$. For a sufficiently small image perturbation Δy,

$$\mathcal{L}_{\text{match}}(y + \Delta y) = \Delta y^\top \mathbb{E}_\zeta \left[J_\zeta(y)^\top J_\zeta(y) \right] \Delta y + O(\|\Delta y\|_2^3). \tag{27}$$

Proof. Taylor expansion gives

$$F_\zeta(y + \Delta y) - F_\zeta(y) = J_\zeta(y)\Delta y + R_\zeta, \qquad \|R_\zeta\|_2 \le c\|\Delta y\|_2^2$$

for a constant c in the local neighborhood. Expanding the squared norm yields the quadratic term in Equation (27), a cross term of order $O(\|\Delta y\|_2^3)$, and a remainder of order $O(\|\Delta y\|_2^4)$. Taking expectations establishes the result.

The matrix $\mathbb{E}_\zeta[J_\zeta^\top J_\zeta]$ measures how visual changes affect the teacher's action predictions. A visual change receives a larger penalty when it causes a larger change in those predictions. Its off-diagonal entries also account for interactions between pixels. The attention visualizations in Appendix A provide a complementary view of the image regions used by AoE; the weighting in Equation (27) is determined by its input derivatives.

Proposition 3: propagation into world-model features. Let $h = H_\theta(\cdot)$ denote world-model features and let $\hat{y} = G_\omega(h)$ denote the predicted future image. We distinguish shared parameters θ from parameters ω specific to the video prediction path. With the teacher fixed,

$$\nabla_h \mathcal{L}_{\text{aux}} = 4\mathbb{E}_\zeta \left[\left(\frac{\partial \hat{y}}{\partial h} \right)^\top J_\zeta(\hat{y})^\top \big(F_\zeta(\hat{y}) - b_\zeta \big) \right], \nabla_\theta \mathcal{L}_{\text{aux}} = \left(\frac{\partial h}{\partial \theta} \right)^\top \nabla_h \mathcal{L}_{\text{aux}}. \tag{28}$$

Proof. Differentiate Equation (25) first with respect to the teacher output, then through the teacher, the image prediction path, and the shared feature extractor. The constant teacher-error term has zero derivative. This gives Equation (28).

Freezing AoE fixes its parameters while retaining the derivative of its output with respect to its input image. Consequently, the auxiliary objective provides a gradient to every trainable component on the generated-image path. The feature derivative can include all shared tensors used by the action head, with those tensors collected into h.

For completeness, suppose the auxiliary objective has an L-Lipschitz gradient in the local parameter neighborhood. An auxiliary step with $g = \nabla_\theta \mathcal{L}_{\text{aux}}$ then satisfies

$$\mathcal{L}_{\text{aux}}(\theta - \eta g) \le \mathcal{L}_{\text{aux}}(\theta) - \eta \left(1 - \frac{L\eta}{2} \right) \|g\|_2^2. \tag{29}$$

This follows by integrating the gradient along the update segment and applying its Lipschitz bound. The auxiliary component therefore decreases its objective for $0 < \eta < 2/L$ whenever $g \neq 0$. Under joint training, the corresponding first-order change also includes the inner product between the native and auxiliary gradients.

Propagation to native action predictions. Let $\hat{\mathbf{a}}_{\theta,\psi} = A_\psi(H_\theta(\cdot))$ be the native action prediction, with ψ denoting parameters specific to the action head. The current observation, instruction, robot state, and native sampling inputs are held fixed. The auxiliary objective depends on the shared features and video prediction path, so $\nabla_\psi \mathcal{L}_{\text{aux}} = 0$. Its contribution to the shared-parameter update changes the native prediction according to

$$\begin{aligned} \hat{\mathbf{a}}_{\theta - \eta g, \psi} - \hat{\mathbf{a}}_{\theta,\psi} &= -\eta \frac{\partial \hat{\mathbf{a}}_{\theta,\psi}}{\partial \theta} g + O(\eta^2) \\ &= -\eta \frac{\partial A_\psi}{\partial h} \frac{\partial h}{\partial \theta} \left(\frac{\partial h}{\partial \theta} \right)^\top \nabla_h \mathcal{L}_{\text{aux}} + O(\eta^2). \end{aligned} \tag{30}$$

The first equality is a Taylor expansion in θ; the second uses Equation (28). This expression isolates the auxiliary contribution to a parameter update. Contributions from the native objective can be added at first order.

The product $(\partial h/\partial\theta)(\partial h/\partial\theta)^\top$ accounts for how parameter updates change the shared features. It also makes explicit that optimizing model parameters and directly optimizing a free feature vector generally produce different updates.

Conditions for transferring an action-error bound. Choose a reference feature vector $h^\star$ whose decoded future satisfies $G_\omega(h^\star) = y$. Define

$$g_\zeta(h) = F_\zeta(G_\omega(h)), \qquad J_\zeta^\star = \left.\frac{\partial g_\zeta}{\partial h}\right|_{h^\star}, \qquad B = \left.\frac{\partial A_\psi}{\partial h}\right|_{h^\star}.$$

We assume the teacher composition and native action map have bounded second derivatives in the neighborhood under consideration.

The transfer condition requires a finite κ such that

$$\|B\Delta h\|_2 \le \kappa \left(\mathbb{E}_\zeta \|J_\zeta^\star \Delta h\|_2^2\right)^{1/2} \tag{31}$$

for the local feature directions being analyzed. This condition says that AoE responds to the feature changes that affect the native action prediction. It permits different output dimensions and coordinate conventions for AoE and the native action head.

An equivalent condition over the full feature space is $\ker M \subseteq \ker B$, where

$$M = \mathbb{E}_\zeta \left[(J_\zeta^\star)^\top J_\zeta^\star\right].$$

To verify the equivalence, first observe that Equation (31) forces $B\Delta h = 0$ whenever $M\Delta h = 0$. Conversely, if $\ker M \subseteq \ker B$, then

$$B = B(M^\dagger)^{1/2} M^{1/2},$$

where $M^\dagger$ inverts the positive eigenvalues of M and is zero on its null space. Taking norms proves Equation (31) with $\kappa = \|B(M^\dagger)^{1/2}\|_{\mathrm{op}}$.

Proposition 4: a local bound on native action error. Let $\mathbf{a}^\star$ be the ground-truth target for the native action prediction, and define its reference error as

$$\epsilon_{\mathrm{head}} = \|A_\psi(h^\star) - \mathbf{a}^\star\|_2.$$

Under the conditions above, there exists a local constant C_h such that

$$\|A_\psi(h) - \mathbf{a}^\star\|_2 \le \epsilon_{\mathrm{head}} + \kappa\left(\sqrt{\frac{\mathcal{L}_{\mathrm{aux}}(h)}{2}} + \frac{\epsilon_{\mathrm{AoE}}}{2}\right) + C_h\|h - h^\star\|_2^2. \tag{32}$$

Proof. Write $\Delta h = h - h^\star$. By Taylor expansion,

$$\begin{aligned} A_\psi(h) - A_\psi(h^\star) &= B\Delta h + R_a, \qquad \|R_a\|_2 \le c_a\|\Delta h\|_2^2, \\ g_\zeta(h) - g_\zeta(h^\star) &= J_\zeta^\star \Delta h + R_{g,\zeta}, \quad \|R_g\|_\zeta \le c_g\|\Delta h\|_2^2. \end{aligned} \tag{33}$$

The sensitivity condition and the triangle inequality give

$$\begin{aligned} \|A_\psi(h) - A_\psi(h^\star)\|_2 &\le \kappa\|J^\star\Delta h\|_\zeta + c_a\|\Delta h\|_2^2 \\ &\le \kappa\|g(h) - g(h^\star)\|_\zeta + (c_a + \kappa c_g)\|\Delta h\|_2^2. \end{aligned} \tag{34}$$

Since $G_\omega(h^\star) = y$, we have $g_\zeta(h^\star) = q_\zeta$. Applying Equation (26) and adding ϵ_{head} proves Equation (32), with $C_h = c_a + \kappa c_g$.

The result connects three quantities: the generated future's fit to the auxiliary target, AoE's accuracy on recorded transitions, and the native head's accuracy at the reference features. For fixed reference quantities and local constants, reducing the auxiliary loss tightens the action-error bound.

Imperfect reference reconstruction and partial coverage. The reference features can also decode an approximation to the recorded future. Define

$$\epsilon_{\mathrm{ref}} = \|g(h^\star) - q\|_\zeta.$$

The same proof adds $\kappa\epsilon_{\mathrm{ref}}$ to Equation (32), because

$$\|g(h) - g(h^\star)\|_\zeta \le \|g(h) - q\|_\zeta + \epsilon_{\mathrm{ref}}.$$

A corresponding extension covers feature directions that AoE does not measure. Let $P = M^\dagger M$ project onto the directions with nonzero teacher sensitivity. Decomposing Δh gives

$$\|B\Delta h\|_2 \le \kappa\|J^\star\Delta h\|_\zeta + \|B(I-P)\Delta h\|_2. \tag{35}$$

Thus the general bound includes $\|B(I-P)(h-h^\star)\|_2$ as an additional term. The auxiliary objective constrains the measured directions, while the native objective also trains action-relevant directions outside that set.

How the action head continues to learn. Shared-feature updates also change the training inputs and gradients of the native action head. Let $g_\psi(\theta,\psi) = \nabla_\psi \mathcal{L}_{\mathrm{native}}(\theta,\psi)$. For an auxiliary change $\Delta\theta = -\eta\nabla_\theta\mathcal{L}_{\mathrm{aux}}$, differentiability gives

$$\begin{aligned} g_\psi(\theta+\Delta\theta,\psi) &= g_\psi(\theta,\psi) + \nabla^2_{\psi\theta}\mathcal{L}_{\mathrm{native}}\,\Delta\theta + o(\|\Delta\theta\|_2) \\ &= g_\psi(\theta,\psi) - \eta\nabla^2_{\psi\theta}\mathcal{L}_{\mathrm{native}}\nabla_\theta\mathcal{L}_{\mathrm{aux}} + o(\eta). \end{aligned} \tag{36}$$

This is the first-order expansion of the native head's gradient. It describes the auxiliary contribution to subsequent native updates: the world model supplies modified features, and the action head learns from those features through its original training objective.

From action-flow discrepancies to sampled actions. The preceding analysis applies to the native prediction map used in the loss. For a flow-based head, a bound on velocity prediction can be propagated through its sampler under trajectory coverage.

To make this step explicit for AoE, let $u_{\hat{y}}(\tau)$ and $u_y(\tau)$ be action trajectories generated under the predicted and recorded futures, starting from the same noise. Suppose the generated-conditioned velocity field is L_f-Lipschitz in the action input. Subtracting the two flow equations yields

$$\frac{d}{d\tau}\|u_{\hat{y}}(\tau) - u_y(\tau)\|_2 \le L_f\|u_{\hat{y}}(\tau) - u_y(\tau)\|_2 + e(\tau),$$

where

$$e(\tau) = \left\|f_\phi(o_t,\hat{y};u_y(\tau),\tau) - f_\phi(o_t,y;u_y(\tau),\tau)\right\|_2.$$

Multiplying by $e^{-L_f\tau}$ and integrating gives

$$\|u_{\hat{y}}(1) - u_y(1)\|_2 \le \int_0^1 e^{L_f(1-\tau)}e(\tau)\,d\tau. \tag{37}$$

If the reference inference-trajectory distribution has density at most C_{cov} times the training noising distribution, then the Cauchy–Schwarz inequality gives

$$\mathbb{E}\|u_{\hat{y}}(1) - u_y(1)\|_2^2 \le C_{\mathrm{cov}}\left(\int_0^1 e^{2L_f(1-\tau)}\,d\tau\right)\mathcal{L}_{\mathrm{match}}. \tag{38}$$

The density condition allows the velocity error along inference trajectories to be controlled by the training expectation. The same argument applies to a native flow head once its velocity discrepancy is bounded. Task success is measured through the execution evaluations in Section 4.

Fixed weights and robust implementations. For positive scalar weights λ_m and λ_g, the combined squared objective has the same form with

$$\begin{aligned} b^{(\lambda)}_\zeta &= \frac{\lambda_m q_\zeta + \lambda_g \mathbf{v}^\star_\zeta}{\lambda_m+\lambda_g}, \\ \lambda_m\mathcal{L}_{\mathrm{match}} + \lambda_g\mathcal{L}_{\mathrm{GT}} &= (\lambda_m+\lambda_g)\|F(\hat{y}) - b^{(\lambda)}\|^2_\zeta + \frac{\lambda_m\lambda_g}{\lambda_m+\lambda_g}\epsilon^2_{\mathrm{AoE}}. \end{aligned} \tag{39}$$

This follows by the same square expansion as Proposition 1.

Fixed coordinate masks and reliability weights restrict the analysis to the supervised coordinates and replace the Euclidean norm with the corresponding weighted norm. For a smooth robust penalty ρ, differentiation produces the same teacher-to-image gradient path, with the residual replaced by its weighted derivative ρ'. Near zero residual, the matching objective has local quadratic form $J^\top W J$, where W contains the coordinate weights and the curvature of ρ at zero.

For the penalty $\rho(r) = \sqrt{1 + r^2} - 1$ used in the implementation,

$$\rho(r) = \frac{r^2}{\sqrt{1 + r^2} + 1}.$$

Therefore, on a bounded residual range $|r| \le R$,

$$\frac{r^2}{\sqrt{1 + R^2} + 1} \le \rho(r) \le \frac{r^2}{2}.$$

With positive weights on the supervised coordinates, the robust and squared objectives consequently bound one another locally. The squared-loss results then carry over with modified constants on those coordinates.

F Training and Evaluation Details

This appendix summarizes the data sources, pretraining configuration, world-model settings, replay protocol, and downstream WAM training configurations used in our experiments. Unless otherwise specified, all settings are kept fixed across the corresponding comparisons.

We separate training provenance from evaluation protocol because the two serve different purposes in the study. The former determines what information a model can use to learn a representation, while the latter determines whether the same representation transfers when task identities, object arrangements, or rendering conditions change. Each table therefore records the source data and optimization choices first, followed by the protocol that is held fixed in the corresponding comparison.

The same convention applies to AoE. A checkpoint is trained once in the canonical-action stage and then reused as a frozen inverse-dynamics model in evaluation and VITA training. Freezing the checkpoint makes the teacher interface identical across world models and WAMs, so differences between rows can be attributed to the image source or the auxiliary objective rather than to a changing action translator. Whenever a table reports replay, the replay environment and initial state are synchronized with the image pair supplied to AoE.

F.1 Canonical-Action Pretraining

AoE is initialized through canonical-action pretraining on a large-scale robot corpus. The corpus contains 9,637 hours of robot interaction data. The authoritative corpus total is reported in Table 9; Table 9 reports the duration and source type of each constituent dataset.

The canonical-action model is trained on 64 H800 GPUs using BF16 precision. We use a global batch size of 1,024 and train for 600K optimization steps. Episodes are sampled with a mixing temperature of 1.0, and each episode provides 1,536 image pairs for canonical-action learning. The remaining optimization and regularization settings are listed in Table 10. Validation is performed on held-out data from DROID, AgibotWorld, and LET.

These validation streams are used for checkpoint selection and calibration only; they do not provide examples from the held-out downstream tasks used in the cross-task tests. The pretraining stage therefore supplies a common camera-space action interface before any target benchmark is introduced. Each training pair is formed from observations that share the same episode context but are separated by a valid interaction interval. Pair sampling preserves the endpoint relation that AoE must recover, while state dropout prevents the model from relying on a single proprioceptive cue when the visual transition already determines the motion. The resulting checkpoint is exported once and reused without parameter updates in all replay, VITA, and WAM comparisons.

Table 9: Data sources for AoE canonical-action pretraining. The reported hours are taken directly from the training configuration.

Data source	Duration (h)	Type
DROID-all-aligned	164.0	Real-robot
AgibotWorld-Beta-all	2174.0	Real-robot
LET-all-aligned	171.0	Real-robot
Xingzhi	260.0	Real-robot
Bodeng-Xinghai	1431.0	Real-robot
Songling	957.0	Real-robot
EgoDex	722.0	Human ego
Groot-live	8.0	Simulation
CALVIN	38.4	Simulation
RMBench	64.5	Simulation
RoboCasa	122.1	Simulation
Intern-A1	3525.0	Simulation
Overall	9637	—

Table 10: AoE canonical-action pretraining configuration.

Category	Setting
Hardware	64 H800 GPUs (8 nodes $\times$ 8 GPUs)
Precision	BF16
Per-GPU batch size	16
Global batch size	1024
Training steps	600K
Warmup steps	3K
Optimizer learning rate	2×10^{-4}
Shuffle buffer	16,384
Pairs sampled per episode	1,536
Dataset mixing temperature	1.0
State dropout	0.5
Visual encoder	Independent DINO
Validation coverage	DROID, AgibotWorld, and LET

The data table records corpus provenance, whereas the configuration table records the optimization contract. Keeping these views separate is useful when reproducing the study: changing the mixture changes the teacher's transfer behavior, while changing the optimizer or sampling schedule changes the quality of the same camera-space target. All downstream rows use the resulting checkpoint under the same frozen-parameter convention.

F.2 World-Model Training and Replay

We compare two Cosmos training configurations. Cosmos + VITA uses the frozen AoE inverse-dynamics model as an action-grounded supervisor and optimizes the VITA objective. Cosmos follows the same model, data, and temporal configuration without IDM supervision or auxiliary VITA losses. The optimizer learning rates follow their respective training configurations. The complete settings are listed in Table 11.

The comparison is made at the level of the supervision path: both models receive the same visual context, temporal window, resolution, hardware budget, and number of optimization iterations, while only the AoE-guided terms are enabled for Cosmos + VITA. This matched setup is important because the replay experiment evaluates the future images produced by the model, rather than a separately trained action policy.

The VITA branch is evaluated only on transitions for which the required image pair and teacher output are available. It does not alter the recorded target video, the native diffusion objective, or the sampling procedure used to produce candidates at test time. At inference, candidates are sampled from the corresponding world-model checkpoint and passed through the same frozen AoE and simulator replay code used for the plain Cosmos condition.

Table 11: World-model training and replay settings. Cosmos + VITA and Cosmos share the same architecture, data, and temporal configuration. Their optimizer learning rates follow the corresponding configurations shown below. Cosmos + VITA additionally uses the frozen AoE teacher and the VITA objective, whereas Cosmos is trained without IDM supervision.

Component	Cosmos + VITA	Cosmos
Hardware	4 nodes $\times$ 8 H800 GPUs	4 nodes $\times$ 8 H800 GPUs
IDM supervision	Frozen AoE teacher	None
Dataloader batch size	4	4
Video resolution	432×768	432×768
Temporal window	93 continuous frames	93 continuous frames
Window stride	92	92
Optimizer learning rate	$2^{-15} = 3.0518 \times 10^{-5}$	$2^{-15} = 3.0518 \times 10^{-5}$
Warmup steps	1,000	1,000
Maximum iterations	10,000	10,000
λ_{match}	1.0	Not applicable
λ_{GT}	1.0	Not applicable
λ_{ROI}	0.1	Not applicable

Table 12: Cross-task world-model replay protocol on RoboTwin. The 50 tasks are divided into two groups of 25 tasks for two-fold cross-task evaluation. Each task is evaluated under clean and randomized configurations.

Category	Setting
Task split	25 training tasks and 25 held-out tasks
Cross-validation	Two folds, with the two task groups exchanged between training and evaluation
Number of task–configuration pairs	100
Replay procedure	AoE predicts actions from image pairs, followed by RoboTwin joint-position replay
IDM replay checkpoint	No additional training for the replay IDM

For Cosmos + VITA, the frozen AoE model provides action-grounded supervision while its parameters remain unchanged. The matching, ground-truth, and ROI terms are combined using the coefficients shown in the table. Cosmos does not use any of these terms.

The table should be read row-wise: the first block defines the common world-model training contract, and the last three rows specify the terms that exist only in the VITA condition. The replay evaluation never fine-tunes AoE on generated frames. Consequently, an image that receives a high execution score must contain enough endpoint and interaction information for the same pretrained translator that handled recorded images.

For RoboTwin, the 50 tasks are divided into two groups of 25 tasks. We use two cross-task folds, with the training and held-out groups exchanged between folds. Each task is evaluated under both clean and randomized configurations, resulting in 100 task–configuration pairs. For each replay condition, AoE predicts actions from image pairs and the predicted actions are converted into RoboTwin joint-position commands for simulator replay. The replay IDM checkpoint is frozen and receives no additional training during evaluation. The complete replay protocol is summarized in Table 12.

The replay rows use a shared reset state for all candidates in a pass@k measurement. This removes the possibility that a successful candidate gains an easier initial configuration than the other candidates. Clean and randomized configurations are rendered with the same task identity, so the cross-task split tests task transfer while the configuration axis tests robustness to scene variation. The detailed task membership is listed later in Table 14.

F.3 Downstream WAM Training

We apply the same frozen AoE teacher to two WAM architectures, LingBot-VA and FastWAM. AoE parameters remain frozen throughout all downstream experiments. The VITA objective is evaluated on both LIBERO and RoboTwin, with the model-specific temporal and input configurations listed in Table 13.

For both LingBot-VA and FastWAM, the LIBERO and RoboTwin models are trained for 5,000 and 50,000 optimization steps, respectively. Table 13 summarizes the hardware, batch sizes, gradient accumulation, optimizer settings, and VITA-specific hyperparameters used in these experiments.

In all WAM experiments, the VITA loss is added during training, while each model retains its native action-generation pipeline for evaluation, including preprocessing, normalization, action decoding, and closed-loop replanning.

Across both architectures and benchmarks, AoE remains frozen, and the matching, GT, and ROI loss coefficients are fixed at 1.0, 1.0, and 0.1, respectively. Batch sizes and optimizer settings are specified separately for each architecture in Table 13. This keeps the auxiliary supervision consistent while accommodating the training configuration of each WAM.

Table 13: WAM training configurations. AoE is frozen in downstream WAM training. RoboTwin uses clean-only training for C → C+R and combined clean/randomized training for C+R → C+R.

	LingBot-VA		**FastWAM**	
Setting	**LIBERO**	**RoboTwin**	**LIBERO**	**RoboTwin**
Hardware	32 H800	32 H800	32 H800	32 H800
Micro batch	4	4	8	8
Gradient accumulation	2	2	2	2
Global batch	256	256	512	512
Training steps	5,000	50,000	5,000	50,000
Learning rate	10^{-5}	10^{-5}	10^{-4}	10^{-4}
Weight decay	0.1	0.1	0.01	0.01
Warmup steps	500	3,000	500	3,000
AoE	Frozen	Frozen	Frozen	Frozen
VITA chunk size	8	8	8	8
VITA interval	2	2	1	1
λ_{match}	1.0	1.0	1.0	1.0
λ_{GT}	1.0	1.0	1.0	1.0
λ_{ROI}	0.1	0.1	0.1	0.1

The two WAM architectures expose different native temporal interfaces, so we report their settings in a single table rather than forcing a common sequence length or action frequency. The shared quantities are the frozen teacher and the VITA coefficients; the model-specific quantities preserve each implementation's original preprocessing and temporal alignment. This is why the same downstream comparison can be read across LIBERO and RoboTwin without interpreting a change in input format as a method change.

F.4 RoboTwin Task Partition

Table 14 lists the complete RoboTwin 25/25 partition. The partition is fixed before training and is shared by the world-model replay and native WAM transfer experiments. In Fold 1, Group 1 is used for training and Group 2 is held out for evaluation. Fold 2 reverses the two groups. This procedure ensures that each task group serves as the held-out task set once.

No task is moved between groups after a model has been trained, and configuration randomization is applied after the task-level assignment. Thus, the held-out fold measures transfer to task identities that are absent from training, while the clean/randomized pairing measures robustness within each identity. Reporting both folds gives every task group one training role and one held-out role.

Table 14: RoboTwin 25/25 task split used for cross-task training and evaluation. The two folds exchange the training and held-out groups: Fold 1 trains on Group 1 and evaluates on Group 2, while Fold 2 reverses the two groups. The same split is used for world-model replay and native WAM transfer.

ID	Group 1	ID	Group 2
1	adjust_bottle	1	blocks_ranking_size
2	beat_block_hammer	2	dump_bin_bigbin
3	blocks_ranking_rgb	3	handover_mic
4	click_alarmclock	4	lift_pot
5	move_can_pot	5	click_bell
6	grab_roller	6	move_pillbottle_pad
7	handover_block	7	move_playingcard_away
8	hanging_mug	8	move_stapler_pad
9	open_laptop	9	place_a2b_right
10	open_microwave	10	place_bread_skillet
11	pick_diverse_bottles	11	place_burger_fries
12	pick_dual_bottles	12	place_cans_plasticbox
13	place_a2b_left	13	place_empty_cup
14	place_bread_basket	14	place_fan
15	place_can_basket	15	place_mouse_pad
16	place_container_plate	16	place_object_basket
17	place_dual_shoes	17	place_object_scale
18	place_object_stand	18	place_phone_stand
19	press_stapler	19	place_shoe
20	rotate_qrcode	20	put_bottles_dustbin
21	scan_object	21	put_object_cabinet
22	shake_bottle	22	shake_bottle_horizontally
23	stack_blocks_two	23	stack_blocks_three
24	stamp_seal	24	stack_bowls_three
25	stack_bowls_two	25	turn_switch

G Detailed experimental results

G.1 Paired execution calibration

Table 15: Calibration of execution and image recovery. The same 300 instances, with successful recorded-action replay and visible grippers, are used in all rows; they cover 60.0% of 500 candidates. Independent geometric recovery reaches 92.0% on real futures.

Input and recovery path	Success (%)
Dataset action replay	**100.0**
GT camera actions	<u>98.0</u>
Dataset image replay + AoE	93.7
VAE-reconstructed dataset image + AoE	91.3
Generated image + AoE	2.0
Generated image + geometry	3.0

The paired set supports comparison of the camera-space adapter, image decoder, VAE reconstruction, and generated-future recovery. The full-population LIBERO table remains the overall performance reference. Calibration coverage and success are separate quantities: the former describes the tested operating range, and the latter describes behavior within it.

G.2 VISUAL-SOURCE DIAGNOSIS, HUMAN JUDGMENTS, AND LOCAL INTERVENTIONS

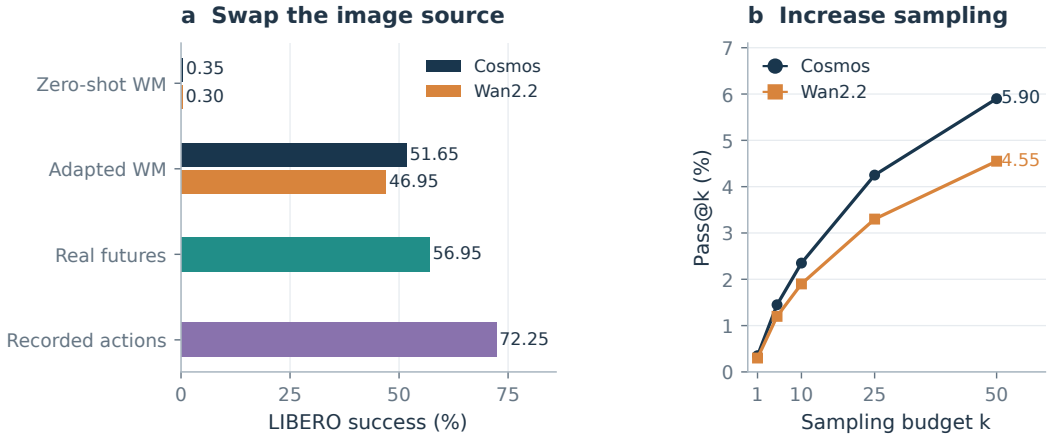


Figure 8: Visual source and sampling budget. Left: episode-weighted LIBERO success with a shared pretrained AoE for image sources; "Adapted WM" denotes the VITA-supervised world model. Right: any-success@k with all candidate executions starting from a common reset state.

A blinded study samples 200 rollouts from each of five future sources and assigns three independent raters to every rollout. Raters see the initial observation, instruction, and video, with model identity and execution outcome hidden. They judge visual plausibility and apparent task completion. Majority vote defines each binary label. Display size, frame rate, and clip duration are shared across sources. Execution is evaluated on the same examples.



Figure 9: Visual plausibility and recovered execution. Human judgments and simulator outcomes on 200 videos per source, with three blinded raters per video. The right panel reports execution conditional on a plausible judgment; "FT" denotes the VITA-supervised condition.

Table 16: Paired human judgments and execution. Each source contains 200 videos judged by three raters. P denotes majority-rated plausibility. All numeric entries are percentages; this subset is separate from the full benchmark evaluations.

Image source	Source	Inference	Plausible	Looks done	Execute	Execute|P	Looks done|Fail
Dataset image	Replay	–	97.0	94.5	57.5	58.8	90.6
Cosmos	Zero-shot	pass@1	77.0	67.5	0.5	0.6	67.3
WAN2.2	Zero-shot	pass@1	75.0	65.0	0.5	0.7	64.8
Cosmos + VITA	Zero-shot	pass@1	89.5	81.0	52.0	56.4	65.6
WAN2.2 + VITA	Zero-shot	pass@1	88.0	77.5	47.0	51.7	61.3

Local geometry and execution tolerance. We use 200 short aligned segments to vary gripper–object geometry by 0, 1, 2, 4, and 8 pixels. Background interventions match the edit area and processing steps. Independent raters score plausibility; action recovery and execution share the original paired reset state. A four-pixel interaction offset reduces execution from 75.5% to 23.0%, while 86.0% of videos retain a plausible judgment. The matched background edit retains 72.5% execution. In a complementary experiment, repairing generated segments gives 3.0% success for background repair, 18.5% for gripper repair, and 32.0% for interaction-region repair, compared with 1.5% before repair. Repairs use aligned ground-truth regions in these short segments.

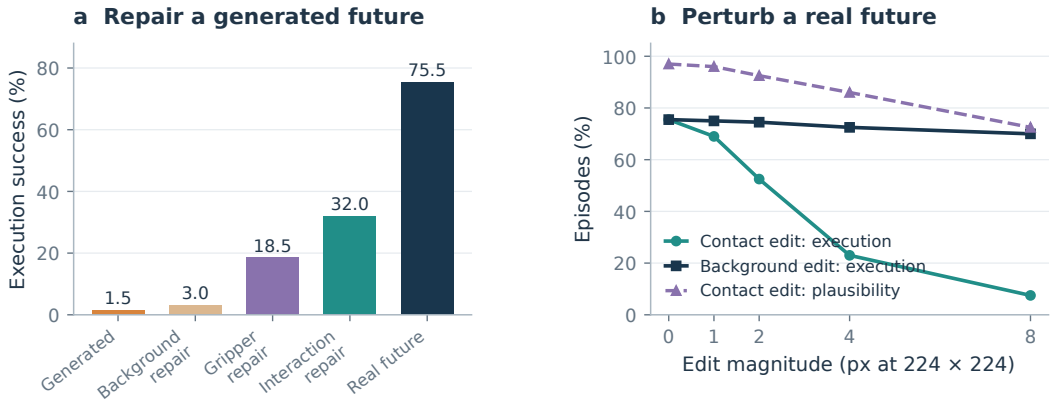


Figure 10: Local image edits and recovered control. Left: execution after repairing selected regions of generated futures. Right: execution after controlled interaction-region or background offsets in recorded futures, together with human-rated plausibility for interaction edits.

Offline metric comparison. For an offline metric comparison, every evaluated pair has a matched reference future. We orient each score so that a larger value predicts better action recovery. SSIM (Wang et al., 2004) and DINO compare the paired images; LPIPS (Zhang et al., 2018) is negated. AoE action agreement is the negative normalized discrepancy between actions recovered from predicted and reference futures. Geometric consistency compares independently recovered end-effector and interaction geometry. Scores are evaluated within tasks before aggregation. The comparison uses held-out episodes for evaluation and a separate source-task partition for selecting score normalization. FVD (Unterthiner et al., 2018) is a distribution-level quantity and is recorded separately at the model/checkpoint level.

Table 17: Offline scores versus execution. AUROC and rank correlation are computed on matched image pairs and aggregated within tasks.

Offline score	Within-task AUROC	Spearman ρ
SSIM	0.55	0.08
DINO feature similarity	0.61	0.17
Negative LPIPS	0.60	0.15
AoE action agreement	<u>0.79</u>	<u>0.49</u>
Geometric consistency	**0.82**	**0.55**

The paired score comparison shows why generic image similarity is not used as the sole selection criterion. SSIM, DINO similarity, and LPIPS summarize appearance agreement, whereas AoE action agreement and geometric consistency use the same endpoint-to-action interface that is ultimately executed. The ranking is computed inside each task before aggregation to prevent tasks with more episodes from dominating the correlation statistics.

G.3 CANDIDATE AND TARGET-DATA BUDGETS

Table 18: Candidate budget on LIBERO. Success (%) over 2,000 initial states, with candidates executed from a shared reset state. Two decimal places retain the precision of this sampling sweep.

Model	IDM action	Source	Inference	Success (%)
Cosmos + AoE	✓	Zero-shot	pass@1	0.35
	✓	Zero-shot	pass@5	1.45
	✓	Zero-shot	pass@10	2.35
	✓	Zero-shot	pass@25	4.25
	✓	Zero-shot	pass@50	5.90
WAN2.2 + AoE	✓	Zero-shot	pass@1	0.30
	✓	Zero-shot	pass@5	1.20
	✓	Zero-shot	pass@10	1.90
	✓	Zero-shot	pass@25	3.30
	✓	Zero-shot	pass@50	4.55

Table 19: Target-task adaptation budget. Adaptation begins from the same-domain source-task checkpoint and uses 500 common evaluation episodes. This target-data sweep is separate from cross-task VITA training. The dataset-image reference is 57.2%.

Model	Target demonstrations	Inference	Success (%)
Cosmos	0	pass@1	3.6
	5	pass@1	18.2
	10	pass@1	29.6
	25	pass@1	43.4
	50	pass@1	54.2
	100	pass@1	55.6
WAN2.2	0	pass@1	3.0
	5	pass@1	15.0
	10	pass@1	25.2
	25	pass@1	38.6
	50	pass@1	50.0
	100	pass@1	53.0

The candidate sweep and target-data sweep answer different questions. The first measures how much can be gained by drawing more futures from one zero-shot checkpoint under a fixed reset. The second measures how quickly a small amount of target-task adaptation closes the gap to the recorded-image reference. Keeping these tables separate avoids treating test-time sampling and training-time adaptation as interchangeable sources of information.

G.4 SUPERVISION OBJECTIVE ABLATIONS

Table 20: Supervision ablation on FastWAM. Held-out success uses LIBERO-90 training and 2,000 LIBERO test episodes; LIBERO-Plus evaluates environment shifts. Native baseline and full-VITA Overall values follow the main-text table.

Training objective	Held-out tasks (%)	LIBERO-Plus (%)
Native baseline	11.2	51.5
Matched-budget native	11.5	50.9
Future DINO alignment	11.8	54.0
AoE smoothness	14.5	51.5
VITA (Match + GT + ROI)	<u>23.6</u>	<u>55.9</u>
Match + GT	**24.1**	**56.4**

The matched-budget native row allocates the additional training compute to the original losses. Future DINO alignment uses the same observed futures and a frozen semantic teacher. The AoE smoothness row uses recovered motion regularity. The full auxiliary objective combines paired teacher matching, confidence-weighted GT flow supervision, and a gripper-centered ROI reconstruction loss. The Match + GT row has zero ROI weight and reaches 24.1% held-out success, compared with 23.6%

Table 22: Teacher-independent evaluation of a VITA-supervised WAN2.2. All rows use the same WAN2.2 checkpoint trained with the AoE-based VITA loss; only the IDM used to recover executable actions at evaluation is changed. The checkmark in "IDM zero-shot" denotes evaluation on tasks unseen during IDM training. J-IDM (Li et al., 2026b) and DreamGen-IDM (Jang et al., 2025) are in-distribution controls and therefore are marked with a cross. Results are success rates in percent.

		LIBERO-Plus								RoboTwin	
IDM evaluator	**IDM zero-shot**	**Background**	**Light**	**Language**	**Noise**	**Layout**	**Camera**	**Robot Init.**	**Overall**	**Easy**	**Hard**
WAN2.2 + AoE	✓	38.5	45.3	37.1	0.1	**39.7**	26.6	13.4	29.9	59.6	40.4
WAN2.2 + J-IDM	×	19.5	31.6	**42.8**	0.0	39.2	**27.7**	–	26.8	**65.5**	**66.7**
WAN2.2 + DreamGen-IDM	×	**39.4**	**46.1**	36.2	**9.4**	35.2	26.1	–	**30.8**	61.9	57.8

for the full objective. This comparison measures the contribution of the local reconstruction term alongside action-flow supervision. Full loss definitions appear in Appendix I. All comparisons use the same source-task partition, frozen pretrained AoE, and native inference path.

The matched-budget row controls for extra optimization, while Future DINO alignment controls for adding a semantic image target without an executable action interface. AoE smoothness controls for a regularizer derived from the teacher without the paired future-image target. Together, these rows isolate the contribution of the action-grounded supervision used by VITA.

G.5 Native WAM control on RoboTwin

Table 21: RoboTwin performance with native WAM inference. C and R denote clean and randomized configurations. Values (%) match the corresponding in-distribution and out-of-distribution main-text tables.

	RoboTwin: C+R → C+R			**RoboTwin: C → C+R**	
Method	**Easy**	**Hard**	**Avg.**	**Easy**	**Hard**
FastWAM	86.4	88.2	87.3	69.5	9.3
FastWAM + VITA	90.7	**91.9**	91.3	73.8	17.5
LingBot-VA	85.6	84.2	84.9	74.3	19.2
LingBot-VA + VITA	**92.4**	91.2	**91.8**	**82.5**	**31.6**

The native WAM table reports the same action decoder and closed-loop evaluator for every row. The first block measures training and testing on the combined clean/randomized distribution, while the second block trains on clean scenes and tests on both clean and randomized scenes. The second block therefore combines task transfer with appearance and layout changes, making the larger VITA gain a direct measure of robustness under a shifted evaluation condition.

H Teacher-independent evaluation of VITA-supervised futures

The main evaluation uses AoE to both supervise the VITA objective and recover actions from generated futures. This appendix tests whether the resulting future videos remain executable when the action-recovery model is changed after training. We keep the WAN2.2 + VITA checkpoint, initial observations, instructions, generated videos, simulator states, and execution code fixed; only the IDM used to translate each future video into an action is replaced. This isolates the dependence of the measured success rate on the evaluator rather than on a change in the world-model sample.

We use J-IDM from VERA (Li et al., 2026b) and the inverse-dynamics model used in DreamGen (Jang et al., 2025). The "IDM zero-shot" column records whether the IDM has been evaluated on a task distribution unseen during its own training. AoE is marked with a checkmark because its canonical-action pretraining is performed independently of the held-out LIBERO-Plus and RoboTwin tasks. The J-IDM and DreamGen-IDM rows are marked with a cross: both models are trained in-distribution for this evaluation and are not used as cross-task teachers. Their scores are therefore teacher-independence controls, while AoE remains the zero-shot cross-task reference.

Recorded-video replay. We first evaluate action recovery from recorded dataset videos to provide a reference for each IDM. Table 23 reports these results alongside recorded-action replay. Over the six evaluated LIBERO-Plus categories, AoE, J-IDM, and DreamGen-IDM achieve 47.0%, 55.5%,

Table 23: Recorded-video replay with different IDM evaluators on LIBERO-Plus. The image-replay rows recover actions from recorded dataset videos; the action-replay row directly executes recorded actions as an oracle reference. Results are success rates in percent. Overall is computed by pooling successful episodes across the six evaluated categories, totaling 8,480 episodes, rather than averaging category-level percentages. Robot Init. is not evaluated in this replay comparison.

Replay setting	Background	Light	Language	Noise	Layout	Camera	Robot Init.	Overall
Dataset action replay (Oracle)	71.9	75.3	78.0	83.8	78.8	67.7	–	76.2
Dataset image replay + AoE	60.7	60.3	59.6	0.0	53.6	57.0	–	47.0
Dataset image replay + J-IDM	65.6	74.5	66.1	0.9	60.1	75.4	–	55.5
Dataset image replay + DreamGen-IDM	66.4	80.0	71.9	3.6	67.2	52.2	–	54.8

Table 24: Teacher-independent evaluation of WAN2.2 futures without VITA supervision on LIBERO-Plus. All rows evaluate futures generated without the VITA loss. AoE is the zero-shot cross-task IDM, while J-IDM and DreamGen-IDM are the in-distribution controls described in Table 22. Results are success rates in percent. Overall pools successful episodes over the evaluated categories: 10,030 episodes across seven categories for AoE, and 8,480 episodes across six categories for J-IDM and DreamGen-IDM. The latter two exclude Robot Init.; their Overall values therefore cover a different evaluation set from AoE's.

IDM evaluator	IDM zero-shot	Background	Light	Language	Noise	Layout	Camera	Robot Init.	Overall
WAN2.2 + AoE	✓	0.9	0.5	0.5	0.0	0.5	0.7	0.3	0.5
WAN2.2 + J-IDM	×	5.5	1.4	2.4	0.4	1.1	0.9	–	1.8
WAN2.2 + DreamGen-IDM	×	2.7	0.7	0.9	0.1	0.5	1.2	–	0.9

and 54.8% success, respectively, compared with 76.2% for recorded-action replay. These results establish that all three IDMs can recover executable actions from recorded visual transitions, while also revealing differences in their recovery performance across perturbations. In particular, sensor noise substantially degrades image-based action recovery for all three IDMs. Recorded-video replay therefore provides a decoder-specific reference rather than assuming that action recovery is equally reliable across all evaluation conditions.

Generated futures without VITA. We next evaluate WAN2.2 futures generated without VITA supervision using the same three IDM evaluators. Table 24 shows that these futures yield low execution success with every decoder: 0.5% with AoE, 1.8% with J-IDM, and 0.9% with DreamGen-IDM. Together with the recorded-video results, this comparison shows that the execution gap between recorded and unsupervised generated futures persists beyond the AoE interface.

VITA gains across IDM evaluators. Comparing Table 24 with Table 22, VITA increases LIBERO-Plus execution success from 0.5% to 29.9% with AoE, from 1.8% to 26.8% with J-IDM, and from 0.9% to 30.8% with DreamGen-IDM. These correspond to improvements of approximately 29.4, 25.0, and 29.9 percentage points, respectively. We compare each evaluator against its own no-VITA baseline; the Overall values should not be interpreted as a ranking across IDMs because their category coverage differs. The improvements with J-IDM and DreamGen-IDM show that the benefit of VITA is not restricted to action recovery by its AoE teacher. Nevertheless, VITA-supervised futures remain below the recorded-video replay reference for each evaluator, leaving a residual gap between generated and recorded visual transitions.

The RoboTwin results in Table 22 provide complementary evidence that VITA-supervised futures remain executable under alternative action-recovery interfaces. J-IDM achieves 65.5%/66.7% on Easy/Hard, and DreamGen-IDM achieves 61.9%/57.8%, compared with 59.6%/40.4% for AoE. Since J-IDM and DreamGen-IDM are trained in-distribution for these evaluations, their results should not be interpreted as zero-shot cross-task generalization by the IDMs themselves. Rather, the LIBERO-Plus comparisons demonstrate improvements under independently trained evaluators, while the RoboTwin results confirm that the VITA-supervised futures support execution beyond the teacher's own action-recovery interface. Together, these controls provide evidence that VITA improves executable visual information rather than merely matching AoE-specific responses.

I COMPLETE TRAINING OBJECTIVE AND GRADIENT IMPLEMENTATION

We detail the implemented LingBot-VA objective and the auxiliary terms in Eq. (53). Native losses preserve each host model's training conventions. Batch indices are suppressed in the auxiliary equations; each per-frame loss is averaged over the minibatch.

1. Native video flow matching. For clean video latent z_0 and Gaussian noise ϵ_z, LingBot-VA constructs

$$z_\sigma = (1-\sigma)z_0 + \sigma\epsilon_z, \qquad v_z^* = \epsilon_z - z_0, \qquad \hat{v}_z = f_\theta(z_\sigma, \sigma, \text{text}, \text{history}). \tag{40}$$

Each latent frame independently samples its diffusion timestep. With B sequences, F latent frames, and N_z spatial-channel entries per frame,

$$\mathcal{L}_{\text{video}} = \frac{1}{BF}\sum_{b,f} w_z(\sigma_{b,f})\frac{\|\hat{v}_{z,b,f} - v^*_{z,b,f}\|_2^2}{N_z}. \tag{41}$$

The scheduler supplies w_z; spatial and channel dimensions are averaged before batch and time.

2. Native action flow matching. The native action tensor follows its corresponding clean-to-noise path,

$$a_{\sigma_a} = (1-\sigma_a)a_0 + \sigma_a\epsilon_a, \qquad v_a^* = \epsilon_a - a_0, \tag{42}$$

$$\mathcal{L}_{\text{native-action}} = \frac{1}{BF}\sum_{b,f} w_a(\sigma_{a,b,f})\frac{\sum_d m_{b,f,d}(\hat{v}_{a,b,f,d} - v^*_{a,b,f,d})^2}{\sum_d m_{b,f,d} + \varepsilon_{\text{den}}}. \tag{43}$$

LingBot-VA retains its 30-channel action tensor. LIBERO supervises the first seven channels, with channel seven using -1 for open and $+1$ for close. The canonical AoE target supplies the auxiliary video losses. FastWAM retains its own native action representation, scheduler, and loss definitions.

3. Canonical action and coordinate weights. For each future time t, $A_t \in \mathbb{R}^{2\times 12}$ contains, per arm, $[u_0, v_0, u_t, v_t, \Delta z_t, R^{6D}_{0\to t}, g_t]$, with normalized absolute UV coordinates and normalized depth displacement. The start-to-future relative rotation occupies six coordinates. Expanding the five group weights $[0,1,1,1,1]$ gives

$$G = (0,0,1,1,1,1,1,1,1,1,1,1). \tag{44}$$

These coordinate weights apply to Match and GT supervision; the complete action tensor remains available in the noisy action input. The arm mask $M_{t,a}$ expresses visibility and label validity for GT supervision and ground-truth gripper ROI construction.

4. Shared AoE action-flow samples. AoE uses $A_{\tau,t} = (1-\tau_t)\xi_t + \tau_t A_t$ and target $V_t^* = A_t - \xi_t$, with uniform τ_t and Gaussian ξ_t. Both branches in Eq. (45) share the same $A_{\tau,t}$, τ_t, and ξ_t. Their common conditions are the window's first image and robot state.

$$P_t = T_\phi(\hat{I}_t; c_t), \qquad R_t = \text{sg}[T_\phi(I_t^{\text{GT}}; c_t)], \tag{45}$$

where c_t collects the shared initial image, robot state, noisy action, and flow time.

5. Match loss. Define the weighted robust reduction

$$\mathcal{H}_w(X,Y) = \frac{\sum_{a,d} w_{t,a,d}\rho_\delta((X_{t,a,d} - Y_{t,a,d})/s_d)}{\sum_{a,d} w_{t,a,d} + \varepsilon_{\text{den}}},$$
$$\rho_\delta(r) = \delta^2\left(\sqrt{1 + (r/\delta)^2} - 1\right). \tag{46}$$

All coordinate scales satisfy $s_d = 1$, and $\delta = 1$. Match uses $w_{t,a,d} = G_d C^{\text{off}}_{t,a,d}$ and compares the predicted-video response P_t with the detached genuine-video response R_t. It is computed whenever the matching term is well-defined, including when the gripper leaves the field of view, and can therefore provide complementary supervision when the corresponding GT supervision is unavailable.

6. GT flow loss and online confidence. GT compares P_t with the analytic velocity V_t^* using weights $MGC^{\text{off}}C^{\text{on}}$. Online confidence is computed from the genuine-video branch:

$$e_{t,a,d} = \text{clip}\left[\left(\frac{R_{t,a,d} - V^*_{t,a,d}}{\eta_d}\right)^2, 0, z^2_{\max}\right], \tag{47}$$

$$C^{\text{on}}_{t,a,d} = \text{clip}\left[\exp\left(-\frac{e_{t,a,d} + |\mathcal{G}(d)|^{-1}\sum_{j\in\mathcal{G}(d)} e_{t,a,j}}{4}\right), c_{\min}, 1\right]. \tag{48}$$

Here $\mathcal{G}(d)$ is the action group containing coordinate d, $c_{\min} = 0.05$, and $z_{\max} = 4$. Tolerances are 0.05 for each start/end UV coordinate, 0.10 for depth, 0.20 for each rotation coordinate, and 0.25 for grip. Confidence is detached and weights supervision according to teacher reliability on the genuine pair.

7. Offline confidence. Calibration cells index data domain, action-flow timestep bin, arm, and coordinate. The five bins are $[0, 0.2)$, $[0.2, 0.4)$, $[0.4, 0.6)$, $[0.6, 0.8)$, and $[0.8, 1]$. For calibration residual r and temperature T_{cal}, the cell's basic confidence is

$$\widetilde{C}^{\text{off}}_c = \exp\left[-\widehat{\mathbb{E}}_c[r^2]/T_{\text{cal}}\right]. \tag{49}$$

Cell sample counts determine shrinkage toward a global conservative fallback. Unavailable domain–timestep cells use that fallback. The resulting table is frozen during WAM training.

8. Ground-truth gripper ROI reconstruction. Let $p_{t,a}$ be the ground-truth future gripper projection in pixel coordinates. We form

$$Q_{t,a}(x,y) = M_{t,a}\exp\left[-\frac{\|(x,y) - p_{t,a}\|_2^2}{2\sigma^2_{\text{px}}}\right], \qquad Q_t(x,y) = \max_a Q_{t,a}(x,y), \tag{50}$$

$$W_t(x,y) = \frac{1 + \alpha Q_t(x,y)}{\text{mean}_{x,y}[1 + \alpha Q_t(x,y)]}, \qquad \alpha = 4, \quad \sigma_{\text{px}} = 10. \tag{51}$$

The normalized weights emphasize the GT gripper endpoints. With C color channels, reconstruction uses

$$\mathcal{L}_{\text{ROI},t} = \frac{\sum_{c,x,y} W_t(x,y)\sqrt{(\hat{I}_{t,c,x,y} - I^{\text{GT}}_{t,c,x,y})^2 + \epsilon^2_{\text{pix}}}}{C\sum_{x,y} W_t(x,y)}, \qquad \epsilon_{\text{pix}} = 10^{-3}. \tag{52}$$

The implemented ROI uses these gripper-endpoint Gaussians. The spatial weighting is computed entirely from ground-truth projections and arm validity.

9. All-future-frame objective. Equation (53) averages Match, GT, and ROI over all $T = 4(F-1)$ genuine future frames at stride one. Their weights are respectively 1, 1, and 0.1. Teacher evaluation uses chunks of eight temporal pairs, retaining every pair and the same temporal average.

$$\mathcal{L}_{\text{aux}} = \frac{1}{T}\sum_{t=1}^{T}\left(\mathcal{L}_{\text{match},t} + \mathcal{L}_{\text{GT},t} + 0.1\mathcal{L}_{\text{ROI},t}\right). \tag{53}$$

10. Optimizer-step schedule. For zero-based optimizer step s, let $\gamma(s) = \min(1, (s+1)/1000)$ and $K = 2$. Define Π_{vs} as the gradient projection onto video-branch and shared parameters. The implemented update uses

$$g_s = \nabla_\theta(\mathcal{L}_{\text{video}} + \mathcal{L}_{\text{native-action}}) + \mathbf{1}[(s+1) \bmod K = 0]\, K\gamma(s)\,\Pi_{\text{vs}}\nabla_\theta\mathcal{L}_{\text{aux}}. \tag{54}$$

Native gradients are evaluated every optimizer step. Each twice step evaluates the complete temporal auxiliary objective and scales its gradient by two, compensating for its evaluation frequency. Warmup reaches full auxiliary weight at the 1,000th optimizer step.

11. Frozen modules and gradient routing. LingBot-VA's native flow definition gives the clean-latent estimate $\hat{z}_0 = z_\sigma - \sigma \hat{v}_z$. The differentiable chain is

$$\hat{v}_z \longrightarrow \hat{z}_0 \longrightarrow \hat{I}_t \longrightarrow T_\phi \longrightarrow \mathcal{L}_{\text{aux}}. \tag{55}$$

The ROI term also contributes a direct image-space gradient through $\hat{I}_t$, bypassing the teacher. AoE and the Wan VAE have frozen parameters. The genuine-video teacher branch is evaluated with gradients disabled. A VAE vector–Jacobian product propagates image-space derivatives to the predicted latents. Auxiliary gradients update video-branch and shared-transformer parameters; action-only parameters receive the native action-flow loss. These routes preserve the native video/action inputs, targets, and inference paths while adding action-aware supervision to video predictions.