RETHINKING REPRESENTATIONS FOR WORLD-ACTION MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

World-action models jointly learn robot policies and predict future observations, making the representation space an interface between control and prediction. We study the design of this space through controlled comparisons, finding that neither reconstruction fidelity nor pre-trained perceptual features alone ensure effective policy learning. These findings motivate ReWAM, a representation-centric world-action model built on pre-trained DINO features. Feature Calibration and a Temporal Representation Bottleneck organize these features into compact world states suited to dynamics modeling. Action-Grounded Representation Shaping routes only action-loss gradients to the bottleneck, thereby letting the policy shape what the representation encodes while the world model learns how it evolves. Without generative video pre-training, ReWAM achieves 93.6% success on RoboTwin 2.0. On RoboDojo, it achieves an average score of 12.29 and a success rate of 8.28% using approximately 600 hours of embodied pre-training data.

1 INTRODUCTION

World-action models (WAMs) jointly learn robot policies and predict how the world evolves. Many recent WAMs (Bi et al., 2026; Li et al., 2026b; Ye et al., 2026c; Yuan et al., 2026b) build on pre-trained video generators and predict future observations in the latent space of a video variational autoencoder (Video-VAE). These models inherit dynamics priors from large-scale generative video pre-training, but their latent spaces are typically optimized for visual reconstruction. Such spaces may preserve appearance details without prioritizing information relevant to control.

The role of world prediction extends beyond producing explicit visual rollouts. Fast-WAM (Yuan et al., 2026b) retains video co-training while removing future-video denoising at deployment, suggesting that prediction can benefit policies through joint training. This raises a fundamental question: **How should the representation space for world-action modeling be defined?**

Representation-space WAMs use pre-trained perceptual features or learned visual-action tokenizers (Lyu et al., 2026; Deng et al., 2026; Wang et al., 2026). Comparative studies show that reconstruction fidelity alone is insufficient to assess control utility (Nilaksh et al., 2026; Yeom et al., 2026). Our experiments support this distinction: without generative video pre-training, raw DINO features outperform Video-VAE latents on RoboTwin 2.0 (Chen et al., 2025), despite lower pixel-reconstruction SSIM. Reconstruction-oriented latents therefore do not by themselves establish effective representations for policy learning.

Perceptual features are also not necessarily well suited to dynamics modeling: frame-wise features do not explicitly encode temporal variation, and their statistics may be poorly matched to diffusion training. Calibrating DINO features through multi-layer aggregation, normalization, and a representation-aware diffusion noise schedule improves average success on RoboTwin 2.0 by 12.20 percentage points. Representation design thus extends beyond the choice of encoder. It must also address how features are prepared for prediction, organized over time, and adapted to the needs of control.

We propose a division of roles: **the policy shapes what the representation encodes, while the world model learns how that representation evolves.** We introduce **ReWAM**, a representation-centric WAM that implements this principle through Action-Grounded Representation Shaping (AGRS). Feature Calibration prepares frozen DINO features for diffusion modeling, and a Temporal

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.

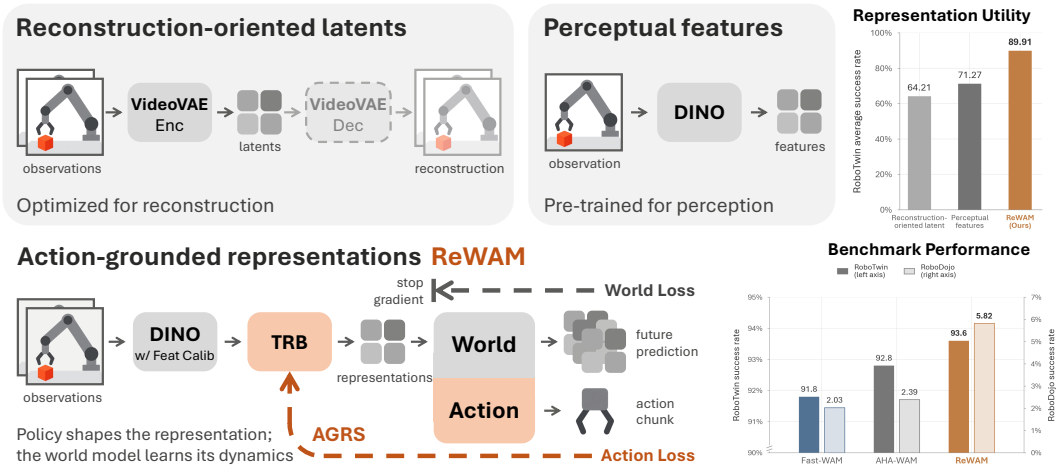

Figure 1: **Overview of representations for world-action modeling.** Top-left: Video-VAE latents are optimized for visual reconstruction. Top-right: Frozen DINO features inherit priors from perceptual pre-training. Bottom: ReWAM uses a Temporal Representation Bottleneck (TRB) to form compact world states from calibrated DINO features. Action-Grounded Representation Shaping (AGRS) trains the TRB through asymmetric routing of action-loss gradients, while the world model learns state transitions.

Representation Bottleneck (TRB) organizes them into compact world states. During joint training, AGRS routes only action-loss gradients to the bottleneck, adapting the representation to action prediction. The world model learns how this representation evolves without updating it through its own prediction loss. Fig. 1 provides an overview of ReWAM.

ReWAM achieves 93.6% success in both the clean and random settings of RoboTwin 2.0. On RoboDojo (Chen et al., 2026b), it leads the compared methods without embodied pre-training, with an average score of 9.42 and a success rate of 5.82%, and embodied pre-training improves these results to 12.29 and 8.28%. Together with real-robot evaluations, these findings highlight a promising path for world-action modeling based on pre-trained perceptual priors and representations shaped by the demands of control, without relying on generative video pre-training.

Our contributions are threefold:

- We study the representation space for WAMs and identify limitations of reconstruction-oriented latents and uncalibrated perceptual features through controlled comparisons.
- We introduce ReWAM, incorporating Feature Calibration, a Temporal Representation Bottleneck, and Action-Grounded Representation Shaping to construct representations for joint prediction and control.
- We evaluate this design through representation ablations, RoboTwin 2.0 and RoboDojo benchmarks, and real-robot experiments, demonstrating its effectiveness without generative video pre-training.

2 Related Work

2.1 World-Action Models

Unified video and action models couple dynamics prediction with policy learning (Zhu et al., 2025; Li et al., 2025b). Recent systems build on video generation through joint denoising (Ye et al., 2026c) or interactions between world and action streams (Bi et al., 2026; Li et al., 2026b; Motubrain Team et al., 2026). RxBrain (Liang et al., 2026) and Cosmos 3 (NVIDIA, 2026) extend this framework to broader multimodal understanding and generation.

Other approaches expose intermediate generative features to the policy (Pai et al., 2026; Ma et al., 2026; Zhang et al., 2026d). Fast-WAM (Yuan et al., 2026b) removes future-video denoising at de-

ployment, while Faster-WAM (Zhao et al., 2026b) studies sparse interactions between video and action prediction under distribution shift. These studies motivate examining how the shared representation interface supports policy learning. Complementary work enriches visual prediction with geometric, motion, and semantic supervision (Guo et al., 2026a; Li et al., 2026c; Yuan et al., 2026a; Li et al., 2026a), or aligns policy features with future visual representations (Zheng et al., 2025b; Zhao et al., 2026a).

Representation-space approaches use pre-trained visual features for prediction and control. LDA-1B (Lyu et al., 2026) and DexWorldModel (Deng et al., 2026) directly predict DINO features, while VLA-JEPA (Sun et al., 2026) and Being-H0.7 (Luo et al., 2026) use future observations to supervise latent representations that guide action generation. LaWAM (Chen et al., 2026a) predicts latent visual subgoals, and GAM (Han et al., 2026) uses geometric foundation features for prediction and action decoding. Learned visual-action tokenizers adapt the interface between perception and control (Wang et al., 2026; Zhang et al., 2026b; Yang et al., 2026), while residual latent actions model changes in DINO features (Zhang et al., 2026c). ReWAM builds on this line of work by calibrating and temporally organizing pre-trained features, then using asymmetric gradient routing to adapt the prediction space to the policy objective during joint world-action training.

2.2 Representations for Prediction and Generation

Predictive learning in visual feature spaces supports representation learning and world modeling, as explored by V-JEPA models (Assran et al., 2025; Mur-Labadia et al., 2026), DINO-world (Baldassarre et al., 2025), and Spa3R (Jiang et al., 2026).

Representation design also matters for generation. REPA (Yu et al., 2025; Leng et al., 2025) and VA-VAE (Yao et al., 2025) align denoiser features and autoencoder latents, respectively, with pre-trained representations. The representation autoencoder (RAE) (Zheng et al., 2026; Singh et al., 2026) uses pre-trained encoders to define generative latent spaces, with subsequent work extending this approach to video generation (Xie et al., 2026; Guo et al., 2026b) and interactive world modeling (Hu et al., 2026).

Probing studies and comparative analyses examine what representations encode and how they support spatial understanding, dynamics prediction, and control (Huang et al., 2026b; Yeom et al., 2026; Nilaksh et al., 2026; Huang et al., 2026a). ReWAM addresses the complementary problem of how to construct and adapt a representation space for joint world prediction and policy learning.

3 ReWAM

3.1 Overview

ReWAM jointly learns an action policy and a predictor of future visual representations. Given the previous and current sampled multi-view observations o_{-1} and o_0, proprioception s_0, and a language instruction ℓ, the policy predicts an action chunk $a_{1:H}$. Here, H denotes the shared future time horizon, covered by actions and visual states at their respective sampling rates. With observations indexed at the image-sampling rate, ReWAM encodes non-overlapping pairs as

$$z_j = E_\phi(o_{j-1}, o_j). \tag{1}$$

The encoder E_ϕ combines a frozen DINOv3-L backbone, feature calibration, and a trainable TRB with parameters ϕ. The current state z_0 conditions both branches, and $z_{1:H}$ provide the world-prediction targets. Sampling and execution details appear in Appendix A.

Following Fast-WAM (Yuan et al., 2026b), ReWAM uses a Mixture-of-Transformers (MoT) architecture with world and action Diffusion Transformer (DiT) branches, denoted by W_θ and A_θ. Both branches receive conditioning embeddings c of ℓ and s_0. Future-state and action tokens attend to current-state world features, but attention between the two token groups is blocked. The world branch uses causal temporal attention and independently sampled noise levels for future states, while the action branch predicts the entire action chunk at once. The conditional model therefore factorizes as

$$p_\theta(z_{1:H}, a_{1:H} \mid z_0, c) = p_\theta(z_{1:H} \mid z_0, c)\, p_\theta(a_{1:H} \mid z_0, c), \tag{2}$$

where θ denotes the trainable WAM parameters. Both branches are trained with flow matching. At inference, the world branch processes z_0 once to condition action generation, without generating future states.

3.2 Feature Calibration

Following RAE (Zheng et al., 2026; Singh et al., 2026), we calibrate DINO representations through multi-layer aggregation, normalization, and a representation-aware noise schedule.

Multi-layer aggregation. For each image and camera view, we extract patch features from DINOv3-L blocks $\mathcal{S} = \{12, 14, 16, 18, 20, 22, 24\}$, using one-based indexing. Let $F^{(l)} \in \mathbb{R}^{N\times d}$ denote the features from block l after non-affine LayerNorm and removal of special tokens, where N is the number of patches in that view and $d = 1024$. We average the selected layers and add the spatial mean of the final layer to every patch:

$$F_p^{\mathrm{MLA}} = \frac{1}{|\mathcal{S}|}\sum_{l\in\mathcal{S}} F_p^{(l)} + \frac{1}{N}\sum_{q=1}^{N} F_q^{(24)}, \qquad p = 1, \ldots, N. \tag{3}$$

This combines features across encoder depths with image-level context while preserving the patch grid and feature width.

Normalization. For the calibrated frozen-feature baseline in tab. 3, we standardize prediction targets using mean and variance estimates fixed before WAM training. In ReWAM, the TRB changes the prediction space during training, so we normalize its outputs with running statistics, as detailed in sec. 3.3.

Representation-aware noise schedule. Let $\tau \in [0, 1]$ denote flow time, with $\tau = 0$ corresponding to clean data and $\tau = 1$ to Gaussian noise. Let D be the total number of scalar elements in the future prediction target, excluding the current state. Following RAE, we sample $u \sim \mathcal{U}(0, 1)$ and set

$$\tau = \frac{\alpha u}{1 + (\alpha - 1)u}, \qquad \alpha = \sqrt{D/D_{\mathrm{base}}}. \tag{4}$$

We set $D_{\mathrm{base}} = 4096$. The shift accounts for both token count and channel width, favoring noisier inputs for larger targets. We sample a separate flow time for each future state using the same shift, while the current state remains clean.

3.3 Temporal Representation Bottleneck

The TRB B_ϕ converts frame-wise DINO features into compact world states. For each camera, we flatten non-overlapping $2\times2\times2$ spatiotemporal blocks into $8d$-dimensional tokens. We concatenate tokens across views, then apply multi-head self-attention and a linear projection to obtain 128-dimensional tokens. Before normalization, the state is

$$z_j^{\mathrm{raw}} = B_\phi(\mathrm{Patchify}\big(F_{j-1}^{\mathrm{MLA}}, F_j^{\mathrm{MLA}}\big)). \tag{5}$$

Causal running normalization. We normalize TRB outputs using per-channel running means μ and variances σ^2 accumulated from preceding batches:

$$z_j = \frac{z_j^{\mathrm{raw}} - \mu}{\sqrt{\sigma^2 + \varepsilon}}, \tag{6}$$

where ε is a numerical stabilizer. We then update μ and σ^2 with the current batch moments using an exponential moving average. The statistics remain fixed at inference.

3.4 Action-Grounded Representation Shaping

AGRS assigns distinct roles to the two objectives: the action objective shapes the TRB, and the future prediction objective trains the world model in the resulting representation space. Only action-loss gradients through the current state z_0 reach the TRB.

Joint world-action learning. We use the linear flow-matching formulation of Fast-WAM. For a clean target y and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, we construct

$$y^\tau = (1-\tau)y + \tau\epsilon. \tag{7}$$

Here, y is a future state or an action chunk, and the target velocity is $\epsilon - y$. At inference, we integrate the predicted action velocity from noise at $\tau = 1$ to data at $\tau = 0$.

Let $z = z_{1:H}$, $a = a_{1:H}$, and $\boldsymbol{\tau}_z = (\tau_{z,1}, \ldots, \tau_{z,H})$. A bar denotes stop-gradient, and noising is applied independently to each future state. The world and action velocity predictors are trained with

$$\mathcal{L}_{\text{world}} = \mathbb{E}\left[\left\|W_\theta(\bar{z}^{\boldsymbol{\tau}_z}, \boldsymbol{\tau}_z; \bar{z}_0, c) - (\epsilon_z - \bar{z})\right\|_2^2\right], \tag{8}$$

$$\mathcal{L}_{\text{action}} = \mathbb{E}\left[\left\|A_\theta(a^{\tau_a}, \tau_a; z_0, c) - (\epsilon_a - a)\right\|_2^2\right]. \tag{9}$$

Asymmetric gradient routing. We jointly train the WAM with both objectives, while routing only action-loss gradients to the TRB:

$$\nabla_\theta \mathcal{L} = \nabla_\theta \mathcal{L}_{\text{world}} + \nabla_\theta \mathcal{L}_{\text{action}}, \qquad \nabla_\phi \mathcal{L} = \left(\frac{\partial z_0}{\partial \phi}\right)^\top \frac{\partial \mathcal{L}_{\text{action}}}{\partial z_0}. \tag{10}$$

World-loss gradients stop at both the current and future representations. Action-loss gradients pass through the world branch's current-state features and z_0 into the TRB. Thus, the world branch learns from both objectives, but the representation encoder is shaped only by the action objective.

4 Experiments

4.1 Setup

Implementation Details. ReWAM uses frozen DINOv3 ViT-L/16 (Siméoni et al., 2026) features and UMT5 embeddings of language instructions. The world and action DiT branches each contain 30 layers, with hidden widths of 2048 and 1024, respectively, and approximately 3.58B parameters in total. Head-camera images are resized to 256×320 pixels and wrist-camera images to 128×160 pixels. Actions and proprioception are represented as 14-dimensional joint-and-gripper vectors and standardized using z-score normalization. We train with AdamW on 32 H20 GPUs, with a batch size of 512 and a peak learning rate of 10^{-4} that decays to 10^{-6} on a cosine schedule.

RoboTwin 2.0. We train a multi-task policy on the 50-task RoboTwin 2.0 benchmark (Chen et al., 2025) using 2,500 clean trajectories and 25,000 trajectories from the random setting. The main comparison uses five training epochs, aligned with the baseline protocol. Representation comparisons and ablations, including the Fast-WAM baselines, use a matched one-epoch training budget. We evaluate 100 episodes per task in each of the clean and random settings and average success rates across tasks.

RoboDojo. We evaluate on the simulation suite of RoboDojo (Chen et al., 2026b), which covers five capability categories: Generalization, Precision, Long-Horizon, Memory, and Open. The training set contains 3,500 trajectories across 35 tasks. We train ReWAM for 50k steps with and without prior embodied pre-training. We report the benchmark's partial-progress score and binary success rate, averaging each metric across the five categories and comparing methods within each pre-training setting.

Embodied Pre-training. To assess the benefit of pre-training on additional embodied data, we pre-train ReWAM and Fast-WAM for 75k steps on the same 600 hours of self-collected real-robot manipulation demonstrations. The dataset covers a broad range of manipulation tasks, with collection environments and camera settings that differ from those used in our downstream real-robot experiments.

Real-Robot Evaluation. We evaluate ReWAM on a bimanual Piper platform across three tasks: Collect Objects, Fold Clothes, and Unpack Lunchbox. After embodied pre-training, ReWAM and Fast-WAM each undergo 10k steps of task-specific post-training on demonstrations collected in the target scene. We report success rates over 20 trials per task.

Table 1: **Evaluation results on RoboTwin 2.0.** Success rates (%) are averaged across 50 tasks, with 100 evaluation episodes per task in each setting. Embodied PT indicates embodied pre-training.

Method	Embodied PT	Clean	Rand	Avg
π_0 (Black et al., 2024)	✓	65.9	58.4	62.2
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	✓	82.7	76.8	79.8
Motus (Bi et al., 2026)	✓	88.7	87.0	87.8
Fast-WAM (Yuan et al., 2026b)	✗	91.9	91.8	91.8
LingBot-VA (Li et al., 2026b)	✓	92.9	91.5	92.2
AHA-WAM (Cai et al., 2026a)	✗	93.4	92.2	92.8
ReWAM (Ours)	✗	**93.6**	**93.6**	**93.6**

Table 2: **Evaluation results on RoboDojo.** SR denotes success rate (%). Fast-WAM* uses the same embodied pre-training as ReWAM. Within each pre-training setting, the best results are in **bold** and the second-best are underlined.

Method	Generalization		Precision		Long-Horizon		Memory		Open		Average	
	Score	SR	Score	SR	Score	SR	Score	SR	Score	SR	Score	SR
w/ Embodied Pre-training												
LDA-1B (Lyu et al., 2026)	0.71	0.17	3.21	0.50	1.92	0.08	2.08	1.78	0.00	0.00	1.58	0.51
InternVLA-A1 (Cai et al., 2026b)	2.87	1.83	3.00	0.92	4.79	1.17	1.58	1.33	0.17	0.17	2.48	1.08
GR00T-N1.7 (NVIDIA et al., 2025)	2.16	1.22	2.54	0.67	8.30	3.58	1.06	0.89	0.18	0.17	2.85	1.31
π_0 (Black et al., 2024)	3.94	2.56	3.56	0.75	6.19	2.00	3.47	2.11	0.25	0.25	3.48	1.53
LingBot-VLA (Wu et al., 2026)	6.71	4.28	5.33	1.83	10.89	5.25	3.82	2.78	0.72	0.67	5.50	2.96
Galaxea-G0-VLA (Jiang et al., 2025)	4.53	2.83	8.10	3.83	12.60	5.58	3.17	1.89	0.70	0.67	5.82	2.96
Xiaomi-Robotics-0 (Cai et al., 2026c)	7.43	5.56	8.42	4.58	13.51	6.92	5.07	3.67	0.22	0.17	6.93	4.18
X-VLA (Zheng et al., 2025a)	10.48	6.78	**18.32**	**12.00**	16.53	9.75	4.76	3.56	0.55	0.50	10.13	6.52
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	13.37	8.17	12.40	5.50	23.54	14.67	5.78	4.56	**1.98**	**1.67**	11.41	6.91
Spatial Forcing (Li et al., 2025a)	14.12	9.33	17.33	10.58	23.26	14.58	5.43	4.11	1.78	1.58	12.38	8.04
HyVLA-0.5 (Zhang et al., 2026a)	11.77	8.39	13.81	8.00	**25.74**	14.92	**13.37**	**12.11**	0.65	0.58	**13.07**	**8.80**
Fast-WAM* (Yuan et al., 2026b)	12.80	9.17	12.32	4.25	22.00	14.50	6.65	5.33	0.95	0.75	10.95	6.80
ReWAM (Ours)	**14.65**	**10.50**	12.32	6.75	25.20	**16.25**	9.05	7.67	0.25	0.25	12.29	8.28
w/o Embodied Pre-training												
Fast-WAM (Yuan et al., 2026b)	2.34	1.11	1.96	0.00	9.14	5.17	3.55	3.44	0.42	0.42	3.48	2.03
AHA-WAM (Cai et al., 2026a)	5.79	3.28	5.86	2.42	8.61	2.67	2.97	2.78	**0.88**	**0.83**	4.82	2.39
GigaWorld-Policy (Ye et al., 2026a)	5.34	2.89	6.15	1.83	15.51	8.92	3.46	2.22	0.54	0.50	6.20	3.27
StarVLA-α (Ye et al., 2026b)	3.93	2.33	9.90	4.33	14.15	6.50	3.34	2.44	0.68	0.58	6.40	3.24
X-WAM (Guo et al., 2026a)	7.39	3.33	6.72	1.83	17.47	9.08	6.32	4.67	0.57	0.25	7.69	3.83
ReWAM (Ours)	**7.52**	**4.33**	**13.30**	**7.50**	**18.90**	**11.00**	**7.00**	**6.00**	0.38	0.25	**9.42**	**5.82**

4.2 Main Results

ReWAM achieves 93.6% success in both settings of RoboTwin 2.0 (tab. 1), exceeding the strongest compared method by 0.2 percentage points on clean and 1.4 points on random. Its performance in the random setting indicates robustness to the benchmark's scene variations without embodied pre-training.

On RoboDojo, ReWAM leads the compared methods without embodied pre-training, with an average score of 9.42 and a success rate of 5.82% (tab. 2). Embodied pre-training raises these results to 12.29 and 8.28%, showing that ReWAM also benefits from additional embodied data pre-training. In this setting, ReWAM achieves the highest Generalization score and success rate and the highest Long-Horizon success rate.

The benefits of embodied pre-training vary across capabilities. Generalization and Long-Horizon success rates rise from 4.33% to 10.50% and from 11.00% to 16.25%, respectively. Precision performance trails methods such as X-VLA and Spatial Forcing, indicating that explicit geometric enhancement can further benefit precision-demanding tasks. ReWAM also falls substantially behind the leading methods in Open instruction following. We attribute the gap in open instruction following to the absence of generative video pre-training, which limits instruction generalization.

Table 3: **Progressive Feature Calibration.** Gen PT denotes generative video pre-training; "+" rows are cumulative.

Model	Gen PT	Clean	Rand
Fast-WAM	✓	87.78	86.43
Fast-WAM	✗	66.10	62.32
Raw DINO WAM	✗	71.00	71.54
+ Normalization		76.10	74.28
+ Repr.-aware noise schedule		77.38	75.82
+ Multi-layer aggregation		**83.90**	**83.03**

Table 4: **Temporal Representation Bottleneck design.**

TRB variant	Clean	Rand
No TRB	83.90	83.03
Fixed random projection	86.76	85.94
RGB recon. codec	71.04	70.48
DINO recon. codec	85.58	86.08
AGRS	**90.70**	**89.12**

Table 5: **AGRS gradient routing.** Checks mark gradients reaching the TRB.

World→TRB		Action→TRB	Clean	Rand
Current	Future	Current		
✓	✓	✓	79.40	77.40
✓	✗	✓	84.32	82.90
✗	✗	✓	**90.70**	**89.12**

Table 6: **TRB dimensionality.**

TRB variant	Dim	Clean	Rand
Fixed random projection	512	86.76	85.94
	256	86.09	85.18
	128	84.26	81.87
AGRS	512	89.08	87.64
	128	**90.70**	**89.12**
	64	89.90	89.08

4.3 Representation Design and Ablations

Reconstruction Fidelity versus Control Utility. Tab. 3 compares Video-VAE and DINO WAMs. Removing generative video pre-training reduces Fast-WAM's success by $21.68/24.11$ percentage points on clean/random. Without this pre-training, raw DINO features outperform Video-VAE latents by $4.90/9.22$ points, despite lower RGB reconstruction SSIM (0.70 versus 0.91). Likewise, an RGB reconstruction codec yields lower success than a DINO reconstruction codec (tab. 4). These comparisons show that reconstruction fidelity does not necessarily ensure a representation's utility for control.

Feature Calibration. As shown in tab. 3, progressively calibrating DINO features with normalization, a representation-aware noise schedule, and multi-layer aggregation yields successive gains of 3.92, 1.41, and 6.87 percentage points in average success. Together, these components improve average success by 12.20 points over raw DINO features. The gains indicate that pre-trained perceptual features benefit from calibration to the distribution and structure required by diffusion-based dynamics modeling.

Temporal Representation Bottleneck. Tab. 4 compares temporal representation designs. Fixed random projection concatenates features at each spatial location across a temporal window and projects them to 512 dimensions. This simple construction improves average success by 2.89 percentage points over the calibrated frame-wise baseline, highlighting the value of temporal organization. The result is consistent with prior work on fixed random projections of pre-trained DINO features as prediction latents (Hu et al., 2026). We also pre-train the TRB as a codec to reconstruct RGB images or DINO features and freeze it during WAM training. The RGB reconstruction codec substantially reduces success, whereas the DINO reconstruction codec falls slightly below fixed random projection on average. Reconstruction-based training thus offers no consistent advantage in constructing temporal representations for control.

Action-Grounded Representation Shaping. AGRS shapes the TRB through the action objective during WAM training. As shown in tab. 4, it achieves $90.70\%/89.12\%$ success on clean/random, exceeding fixed random projection by $3.94/3.18$ percentage points. Tab. 5 isolates the effect of gradient routing. Allowing world-loss gradients through the current-representation path reduces average success by 6.30 points, while allowing them through both current and future paths further reduces success to $79.40\%/77.40\%$. These results suggest that allowing the world objective to reshape its own prediction targets may encourage representation collapse or predictive shortcuts. Tab. 6 exam-

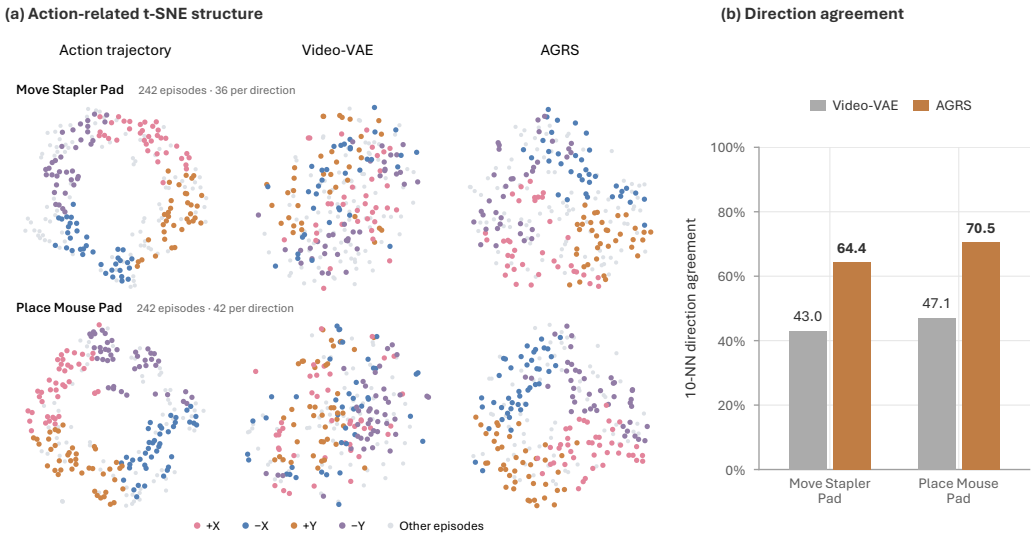

Figure 2: **Action-related structure in representation variations.** (a) Independent t-SNE projections of commanded end-effector trajectories, Video-VAE variations, and AGRS variations (left to right) for Move Stapler Pad (top) and Place Mouse Pad (bottom). Each task shows the same 242 transitions in all three projections; colors identify motion directions in balanced subsets, with the remaining transitions in gray. (b) Mean direction agreement among ten nearest neighbors in the original representation space. AGRS shows higher agreement than Video-VAE on both tasks.

ines representation dimensionality. Reducing fixed random projection from 512 to 128 dimensions lowers average success by 3.29 percentage points. AGRS performs best among the tested widths at 128 dimensions and remains effective at 64 dimensions. These results suggest that action grounding concentrates control-relevant information, supporting effective control at substantially lower dimensionality.

4.4 Action-Related Structure in Representations

We investigate the action-related structure in ReWAM and Video-VAE representations using t-SNE and nearest-neighbor direction agreement. Fig. 2(a) shows independent t-SNE projections of commanded end-effector trajectories and representation variations for the same transitions: one from each of 242 left-arm episodes per task, selected from demonstrations in the random setting. Colors indicate the dominant direction of commanded displacement along the robot's coordinate axes. Highlighted subsets are balanced across directions by displacement magnitude, while the remaining transitions appear in gray. AGRS exhibits clearer local grouping by direction, suggesting that its representation variations capture action-related information.

We also measure whether nearby representation variations correspond to the same motion direction in the original representation space. We find each highlighted transition's ten nearest neighbors within the balanced subset, excluding itself, and compute the fraction sharing its commanded-motion direction. Fig. 2(b) reports the mean agreement, which is higher in AGRS space than in Video-VAE space.

4.5 Real-Robot Evaluation

We assess ReWAM on physical hardware across three tasks. Collect Objects requires picking up tabletop items and placing them in a basket; Fold Clothes involves manipulating a garment; and Unpack Lunchbox requires opening a lunchbox and retrieving food. As shown in fig. 3, ReWAM achieves success rates of 100%, 85%, and 90%, respectively, compared with 100%, 70%, and 70% for Fast-WAM. Both methods complete all 20 Collect Objects trials, while ReWAM gains 15 percentage points on Fold Clothes and 20 points on Unpack Lunchbox. Overall, ReWAM succeeds in

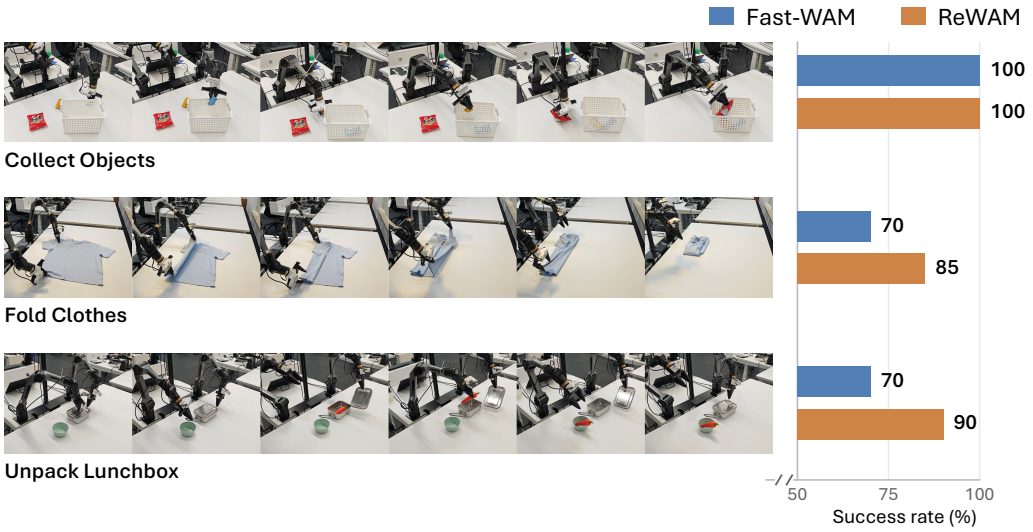

Figure 3: **Real-robot task success.** Task photos and success rates over 20 trials per task. ReWAM achieves 91.7% average success versus 80.0% for Fast-WAM. The horizontal axis starts at 50%.

55 of 60 trials versus 48 of 60 for Fast-WAM, raising average success from 80.0% to 91.7%. The gains extend to the evaluated deformable-object and multi-stage manipulation tasks. We also observe autonomous retries, despite their absence from the task demonstrations. Additional real-robot demonstrations are available in the supplementary materials.

5 Conclusion

Our study shows that effective world-action modeling depends on how its representation space is constructed and adapted. Reconstruction fidelity alone does not determine control performance, and pre-trained perceptual features benefit from calibration and temporal organization. ReWAM combines these components with action-grounded representation shaping, letting the policy determine what the representation encodes, while the world model learns its dynamics. Results on RoboTwin 2.0, RoboDojo, and real-robot manipulation demonstrate the effectiveness of this design without generative video pre-training. These findings highlight the potential of representation design to advance joint prediction and control.

AI use statement

Codex and DeepSeek assisted with research discussions, implementation, experimental analysis, and manuscript drafting and revision. The authors have manually reviewed all AI-assisted work and take responsibility for its correctness, soundness, and fairness.

Reproducibility statement

The Experiments section and appendix describe the experimental setup and evaluation protocols. To support reproduction and further research, we will publicly release the training and deployment code, model weights, configurations, and data no later than publication of the accepted paper.

References

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong

Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Federico Baldassarre, Marc Szafraniec, Basile Terver, Vasil Khalidov, Francisco Massa, Yann LeCun, Patrick Labatut, Maximilian Seitzer, and Piotr Bojanowski. Back to the features: DINO as a foundation for video world models. *arXiv preprint arXiv:2507.19468*, 2025.

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

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Jisong Cai, Long Ling, Shiwei Chu, Zhongshan Liu, Jiayue Kang, Zhixuan Liang, Wenjie Xu, Yinan Mao, Weinan Zhang, Xiaokang Yang, Ru Ying, Ran Zheng, and Yao Mu. AHA-WAM: Asynchronous horizon-adaptive world-action modeling with observation-guided context routing. *arXiv preprint arXiv:2606.09811*, 2026a.

Junhao Cai, Zetao Cai, Jiafei Cao, Yilun Chen, Zeyu He, Lei Jiang, Hang Li, Hengjie Li, Yang Li, Yufei Liu, Yanan Lu, Qi Lv, Haoxiang Ma, Jiangmiao Pang, Yu Qiao, Zherui Qiu, Yanqing Shen, Xu Shi, Yang Tian, Bolun Wang, Hanqing Wang, Jiaheng Wang, Tai Wang, Xueyuan Wei, Chao Wu, Yiman Xie, Boyang Xing, Yuqiang Yang, Yuyin Yang, Qiaojun Yu, Feng Yuan, Jia Zeng, Jingjing Zhang, Shenghan Zhang, Shi Zhang, Zhuoma Zhaxi, Bowen Zhou, Yuanzhen Zhou, Yunsong Zhou, Hongrui Zhu, Yangkun Zhu, and Yuchen Zhu. InternVLA-A1: Unifying understanding, generation and action for robotic manipulation. *arXiv preprint arXiv:2601.02456*, 2026b.

Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, Feng Qiu, Heng Qu, Yifei Su, Qiao Sun, Dong Wang, Donghao Wang, Yunhong Wang, Rujie Wu, Diyun Xiang, Yu Yang, Hangjun Ye, Yuan Zhang, and Quanyun Zhou. Xiaomi-Robotics-0: An open-sourced vision-language-action model with real-time execution. *arXiv preprint arXiv:2602.12684*, 2026c.

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, Yuanbo Xu, and Chao Yu. LaWAM: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026a.

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.

Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Haoran Lu, Weijie Wan, Baijun Chen, Songling Liu, Haowen Yan, Honghao Su, Zhiyang Dou, Kaixuan Wang, Dandan Zhang, Yunze Liu, Yan Qin, Qiwei Liang, Qiwei Wu, Zijian Lin, Wenwei Lin, Yuran Wang, Minghua He, Tianshu Wu, Ruihai Wu, Jingquan Zhou, Kai-Chong Lei, Haibao Yu, Yuanfeng Ji, Weiyang Jin, Guanyu Lin, Xiaofan Li, Qi Xiong, Renjing Xu, Zhongyu Li, Wenhao Chai, Enze Xie, Ziwei Wang, Yao Mu, Hao Dong, Wojciech Matusik, Mingyu Ding, Wenbo Ding, Ping Luo, and Masayoshi Tomizuka. RoboDojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026b.

Yueci Deng, Guiliang Liu, and Kui Jia. DexWorldModel: Causal latent world modeling towards automated learning of embodied tasks. *arXiv preprint arXiv:2604.16484*, 2026.

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

Minghui Guo, Shengqiong Wu, and Hao Fei. V-RAE: Rethinking video latent spaces for generation. *arXiv preprint arXiv:2608.13556*, 2026b.

Jisang Han, Seonghu Jeon, Jaewoo Jung, René Zurbrügg, Honggyu An, Tifanny Portela, Marco Hutter, Marc Pollefeys, Seungryong Kim, and Sunghwan Hong. Geometric action model for robot policy learning. *arXiv preprint arXiv:2606.17046*, 2026.

Anthony Hu, Václav Volhejn, Adrien Ramanana Rahary, Chris Mulder, Aditya Makkar, Alyx Liao, Amélie Royer, Manu Orsini, Adam Jelley, Eloi Alonso, Florian Laurent, Fredrik Norén, James Swingos, Jan Hünermann, Kent Rollins, Lucas Hosseini, Matthieu Le Cauchois, Maxim Peter, Pim de Witte, Tim Brown, Vincent Micheli, Moritz Böhle, Gabriel de Marmiesse, Viktoriia Sharmanska, Lucia Specia, Michael Black, and Patrick Pérez. Multiplayer interactive world models with representation autoencoders. *arXiv preprint arXiv:2607.05352*, 2026.

Siqiao Huang, Partha Kaushik, Michael Chen, Hengkai Pan, Kaiwen Geng, Omar Chehab, Fernando Moreno-Pino, and Max Simchowitz. Nano world models: A minimalist implementation of future video prediction. *arXiv preprint arXiv:2605.23993*, 2026a.

Zixuan Huang, Xiang Li, Zhaoyang Lv, and James M. Rehg. How much 3D do video foundation models encode? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 384–394, 2026b.

Haoyi Jiang, Liu Liu, Xinjie Wang, Yonghao He, Wei Sui, Zhizhong Su, Wenyu Liu, and Xinggang Wang. Spa3R: Predictive spatial field modeling for 3D visual reasoning. *arXiv preprint arXiv:2602.21186*, 2026.

Tao Jiang, Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Jianning Cui, Xiao Liu, Shuiqi Cheng, Jiyang Gao, Huazhe Xu, and Hang Zhao. Galaxea open-world dataset and G0 dual-system VLA model. *arXiv preprint arXiv:2509.00576*, 2025.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-E: unlocking VAE for end-to-end tuning with latent diffusion transformers. In *IEEE/CVF International Conference on Computer Vision*, pp. 18262–18272, 2025.

Baoyu Li, Xinchen Yin, Mengying Lin, Yixin Zhang, and Danfei Xu. EgoWAM: World action models beyond pixels with in-the-wild egocentric human data. *arXiv preprint arXiv:2607.08436*, 2026a.

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

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Proceedings of Robotics: Science and Systems*, 2026b.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified Video Action Model. In *Proceedings of Robotics: Science and Systems*, 2025b.

Ying Li, Xiaobao Wei, Jiajun Cao, Hao Wang, Xiaowei Chi, Chengyu Bai, Qianpu Sun, Jiajun Li, Xiaojie Zhang, Peidong Jia, Jian Tang, Sirui Han, and Shanghang Zhang. WAM4D: Fast 4D world action model via spatial register tokens. *arXiv preprint arXiv:2606.14048*, 2026c.

Haotian Liang, Mingkang Chen, Yufei Huang, Yuchun Guo, Xiaomeng Zhu, Xiangli Shi, Kaixuan Wang, Yunxuan Mao, Weijie Zhou, Ling Chen, Shirong Zeng, Yueyu Long, Yuchen Si, Yajuan Zhu, Xingyu Zhou, Minghui Wang, Wanjia He, Xin Yang, Lingzhu Xiang, Zhiqing Liu, Bohan Ma, Xiran Huang, Tianshuo Yang, Zhiheng Liu, Xuantang Xiong, Zisheng Lu, Ping Luo, Yao Mu, Han Hu, and Zhengyou Zhang. RxBrain: Embodied cognition foundation model with joint language-visual reasoning and imagination. *arXiv preprint arXiv:2607.14187*, 2026.

Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-H0.7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026.

Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, Wenbo Cui, Senmao Qi, Shuo Wang, Yixin Zheng, Mi Yan, Xuesong Shi, Haoran Li, Dongbin Zhao, Ming-Yu Liu, Zhizheng Zhang, Li Yi, Yizhou Wang, and He Wang. LDA-1B: Scaling latent dynamics action model via universal embodied data ingestion. In *Proceedings of Robotics: Science and Systems*, 2026.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. DiT4DiT: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026.

Motubrain Team, Chendong Xiang, Fan Bao, Haitian Liu, Hengkai Tan, Hongzhe Bi, James Li, Jiabao Liu, Jingrui Pang, Kiro Jing, Louis Liu, Mengchen Cai, Rongxu Cui, Ruowen Zhao, Runqing Wang, Shuhe Huang, Yao Feng, Yinze Rong, Zeyuan Wang, and Jun Zhu. Motubrain: An advanced world action model for robot control. *arXiv preprint arXiv:2604.27792*, 2026.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

Nilaksh, Saurav Jha, Artem Zholus, and Sarath Chandar. Reconstruction or semantics? What makes a latent space useful for robotic world models. *arXiv preprint arXiv:2605.06388*, 2026.

NVIDIA. Cosmos 3: Omnimodal world models for physical AI. *arXiv preprint arXiv:2606.02800*, 2026.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Jonas Pai, Liam Achenbach, Oliver Sanchez, Stefanos Charalambous, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond VLAs. In *Proceedings of Robotics: Science and Systems*, 2026.

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

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun 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. *Transactions on Machine Learning Research*, 2026.

Jaskirat Singh, Boyang Zheng, Zongze Wu, Richard Zhang, Eli Shechtman, and Saining Xie. Improved baselines with representation autoencoders. *arXiv preprint arXiv:2605.18324*, 2026.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: Enhancing vision-language-action model with latent world model. In *European Conference on Computer Vision*, 2026.

Junke Wang, Qihang Zhang, Shuai Yang, Yiming Luo, Yujun Shen, Zuxuan Wu, Yu-Gang Jiang, and Yinghao Xu. RepWAM: World action modeling with representation visual-action tokenizers. *arXiv preprint arXiv:2606.13674*, 2026.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, Yiyu Ren, Kejia Zhang, Hui Yu, Jingmei Zhao, Shuai Zhou, Zhenqi Qiu, Houlong Xiong, Ziyu Wang, Zechen Wang, Ran Cheng, Yong-Lu Li, Yongtao Huang, Xing Zhu, Yujun Shen, and Kecheng Zheng. A pragmatic VLA foundation model. *arXiv preprint arXiv:2601.18692*, 2026.

Zhihao Xie, Junfeng Wu, Xinting Hu, Junchao Huang, and Li Jiang. VideoRAE: Taming video foundation models for generative modeling via representation autoencoders. *arXiv preprint arXiv:2607.14088*, 2026.

Fan Yang, Yuting Su, Xiaobo Wang, Yuncheng You, Fugui Fan, Yuting Wu, Minghui Wu, Chenxu Zhao, JiaHong Ning, and Peiguang Jing. LiLa-WAM: Lightweight latent reasoning world-action model for robotic manipulation. *arXiv preprint arXiv:2608.03701*, 2026.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15703–15712, 2025.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, Min Cao, Peng Li, Qiuping Deng, Wenjun Mei, Xiaofeng Wang, Xinze Chen, Xinyu Zhou, Yang Wang, Yifan Chang, Yifan Li, Yukun Zhou, Yun Ye, Zhichao Liu, and Zheng Zhu. GigaWorld-Policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026a.

Jinhui Ye, Ning Gao, Senqiao Yang, Jinliang Zheng, Zixuan Wang, Yuxin Chen, Pengguang Chen, Yilun Chen, Shu Liu, and Jiaya Jia. StarVLA-α: Reducing complexity in vision-language-action systems. *arXiv preprint arXiv:2604.11757*, 2026b.

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

Jewon Yeom, Hanseul Kim, Jeongjae Park, Sungmok Jung, Jaejin Lee, and Taesup Kim. What makes video world model latents action-relevant: Prediction over reconstruction. *arXiv preprint arXiv:2606.07687*, 2026.

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

Shanglin Yuan, Weiheng Zhao, Xin Shi, Haoyi Jiang, Xianda Guo, Liu Liu, Wenyu Liu, Wei Sui, and Xinggang Wang. DreamWAM: Beyond RGB future prediction for world action models. *arXiv preprint arXiv:2608.04996*, 2026a.

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*, 2026b.

He Zhang, Lingzhu Xiang, Haitao Lin, Zeyu Huang, Minghui Wang, Dingyan Zhong, Yubo Dong, Yihao Wu, Yongming Rao, Dongsheng Zhang, Wanjia He, Ling Chen, Kai Huang, Jiahao Chen, Sichang Su, Xumin Yu, Ziyi Wang, Chengwei Zhu, Xiao Teng, Yuchun Guo, Yufeng Zhang,

Yuandong Liu, Rui Wang, Zisheng Lu, Han Hu, and Zhengyou Zhang. Hy-Embodied-0.5-VLA: From vision-language-action models to a real-world robot learning stack. *arXiv preprint arXiv:2606.14409*, 2026a.

Qihang Zhang, Lin Li, Luyao Zhang, Shuai Yang, Yiming Luo, Shuaiting Li, Ruilin Wang, Junke Wang, Jiahao Shao, Gangwei Xu, Jiaming Zhou, Yishu Shen, Yudong Jin, Fangyi Xu, Shuailei Ma, Jiaqi Liao, Guanxing Lu, Zifan Shi, Yongkun Wen, Yujie Zhao, Weixuan Tang, Xinyang Wang, Chaojian Li, Jiapeng Zhu, Ka Leong Cheng, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Native video-action pretraining for generalizable robot control. *arXiv preprint arXiv:2607.08639*, 2026b.

Xinyu Zhang, Zhengtong Xu, Yutian Tao, Yeping Wang, Yu She, and Abdeslam Boularias. Learning visual feature-based world models via residual latent action. *arXiv preprint arXiv:2605.07079*, 2026c.

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

Han Zhao, Jingbo Wang, Wenxuan Song, Shuai Chen, Yang Liu, Yan Wang, Haoang Li, and Donglin Wang. FRAPPE: Infusing world modeling into generalist policies via multiple future representation alignment. *arXiv preprint arXiv:2602.17259*, 2026a.

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

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, 2026.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025a.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loïc Magne, Avnish Narayan, You Liang Tan, Guanzhi Wang, Qi Wang, Jiannan Xiang, Yinzhen Xu, Seonghyeon Ye, Jan Kautz, Furong Huang, Yuke Zhu, and Linxi Fan. FLARE: Robot learning with implicit world modeling. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3952–3971. PMLR, 2025b.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified World Models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Proceedings of Robotics: Science and Systems*, 2025.

A Implementation and Evaluation Details

Temporal sampling and horizons. Each training clip contains ten multi-view observations sampled every 8 control steps, at offsets $[-8, 0, 8, 16, 24, 32, 40, 48, 56, 64]$. The first pair defines z_0, and the remaining four pairs define $z_{1:4}$ over the 64-action horizon. Inference requires only the historical and current observations. At the first query, the current observation substitutes for the missing history. We apply the same substitution during RoboTwin training with probability 0.1. Padded targets are excluded from the corresponding losses.

TRB architecture. The head camera yields a 16×20 DINO patch grid, and each wrist camera yields an 8×10 grid. Spatiotemporal patchification produces $80 + 20 + 20 = 120$ tokens per state, each initially 8192-dimensional. A single eight-head attention layer mixes tokens across views within each temporal pair. It uses two-dimensional RoPE with base 10,000, coordinates scaled by 16, and learned camera-specific query and key biases.

Inference and action execution. At each query, we cache the world branch's current-state keys and values, generate actions from Gaussian noise using Euler updates with decreasing flow time, and execute a prefix of the predicted chunk. Tab. 7 lists the execution and denoising budgets used for evaluation.

Table 7: **ReWAM inference settings.** Executed actions are the prefix applied before the next policy query.

Setting	Predicted actions	Executed actions	Denoising steps
RoboTwin 2.0	64	24	10
RoboDojo	64	10	20
Piper real robot	64	32	20

B Embodied Pre-training Data Scale

Tab. 8 compares the reported durations of embodied pre-training datasets for selected methods evaluated on RoboDojo. ReWAM uses approximately 600 hours of self-collected robot demonstrations, substantially less than the amounts reported for the listed baselines. With this pre-training, ReWAM achieves an average score of 12.29 and a success rate of 8.28%, as shown in tab. 2. Our matched Fast-WAM baseline uses the same 600-hour dataset. Differences in data composition should be considered when interpreting the comparison.

Table 8: **Embodied pre-training data scale for selected RoboDojo methods.** Hours measure the duration of embodied training data. A "+" indicates a reported lower bound, and $\approx$ denotes an approximate duration.

Method	Data duration in hours
Galaxea-G0-VLA (Jiang et al., 2025)	2,000+
π_0 (Black et al., 2024)	10,000+
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	10,000+
HyVLA-0.5 (Zhang et al., 2026a)	10,000+
LingBot-VLA (Wu et al., 2026)	$\approx$ 20,000
ReWAM	$\approx$ 600

C Contribution of World Co-training

To isolate the contribution of world co-training, we disable the world loss in the raw DINO WAM configuration from tab. 3 while keeping the backbone and architecture unchanged. As shown in tab. 9, world co-training improves clean/random success by 2.60/2.99 percentage points, providing gains beyond those from pre-trained DINO features alone.

Table 9: **World-loss ablation for DINO WAM on RoboTwin 2.0.**

Method	Clean	Rand
DINO policy	68.40	68.55
Raw DINO WAM	71.00	71.54

D Detached Pixel-Reconstruction Probe

The SSIM probe measures how much pixel information can be recovered from the representation. During training, an auxiliary decoder $D_\psi(\bar{z}_j)$ reconstructs future RGB observation pairs from detached latents. The reconstruction loss updates only the decoder parameters ψ, without affecting the encoder or WAM. The decoder is not used at inference.

E Representation Analysis Protocol

Data source and transition selection. We examine all 500 RoboTwin demonstrations from the random setting for each displayed task. We retain episodes in which the left gripper's command range exceeds 0.1 and the right gripper's does not. Candidate transitions start at $16, 24, 32, \ldots$ and span 32 control steps. Throughout each window, including both endpoints, the active gripper's command and recorded state must remain below 0.2 and 0.25, respectively, in dataset units. We select the middle eligible start in each episode, using the later of the two middle starts when the count is even. This yields 242 transitions each for Move Stapler Pad and Place Mouse Pad. Selection uses only action and state records, independently of representation features or projections.

Direction balancing. We assign each transition to the signed axis with the largest absolute endpoint displacement. To qualify for highlighting, a transition must have a displacement of at least 0.04 m within 45° of that axis. Directions with fewer than 20 eligible candidates are excluded. Within displacement-magnitude bins $[0.10, 0.14)$, $[0.14, 0.18)$, $[0.18, 0.22)$, and $[0.22, 0.30)$ m, we sample equal counts per direction without replacement. We omit any bin with fewer than three candidates in a retained direction. This yields 144 highlighted transitions for Move Stapler Pad and 168 for Place Mouse Pad, with 36 and 42 per direction, respectively. Both subsets cover $+X$, $-X$, $+Y$, and $-Y$.

t-SNE projection. Each t-SNE fit includes all 242 transitions, with those outside the highlighted subset shown in gray. We use Euclidean distances, perplexity 30, PCA initialization, an automatic learning rate, and 1500 iterations. We do not reduce feature dimensionality with PCA or standardize features before fitting. Projections are fitted independently, so orientations and absolute positions cannot be compared across panels.