Mila

# Reconstruction or Semantics? What Makes a Latent Space Useful for Robotic World Models

**Nilaksh**[*1,2,3] **Saurav Jha**[*1,2,3] **Artem Zholus**[*1,2,3] **Sarath Chandar**[1,2,3,4]

[1]Chandar Research Lab [2]Mila – Quebec AI Institute [3]Polytechnique Montréal [4]Canada CIFAR AI Chair

[*]Equal Contribution
Correspondence: [nilaksh.nilaksh, saurav.jha]@mila.quebec

https://hskalin.github.io/semantic-wm/
https://huggingface.co/Nilaksh404/semantic-wm



**Abstract** World model-based policy evaluation is a practical proxy for testing real-world robot control by rolling out candidate actions in action-conditioned video diffusion models. As these models increasingly adopt latent diffusion modeling (LDM), choosing the *right latent space* becomes critical. While the status quo uses autoencoding latent spaces like VAEs that are primarily trained for pixel *reconstruction*, recent work suggests benefits from pretrained encoders with representation-aligned *semantic* latent spaces. We systematically evaluate these latent spaces for action-conditioned LDM by comparing six reconstruction and semantic encoders to train world model variants under a fixed protocol on BridgeV2 dataset, and show effective world model training in high-dimensional representation spaces with and without dimension compression. We then propose three axes to assess robotic world model performance: visual fidelity, planning and downstream policy performance, and latent representation quality. Our results show visual fidelity alone is insufficient for world model selection. While reconstruction encoders like VAE and Cosmos achieve strong pixel-level scores, semantic encoders such as V-JEPA 2.1 (strongest overall on policy), Web-DINO, and SigLIP 2 generally excel across the other two axes at all model scales. Our study advocates semantic latent space as stronger foundation for policy-relevant robotics diffusion world models.

## 1 Introduction

Action-conditioned video world models are emerging as a practical interface between generative modeling and robotics [20, 70, 10]. Given observation and action histories, they predict future observations and serve as learned proxies for robot-environment interaction when handcrafted simulators are difficult to build [58, 15]. Recent works show that such models can support policy evaluation with good correlation to real-world outcomes [62], and policy improvement [82, 75, 52]. Yet current evaluations say little about which representation makes a world model faithful to robotic dynamics.

This question is increasingly important because many video world models are latent diffusion models (LDMs) [64, 48] that learn dynamics in an encoder-defined latent space. The standard choice is a reconstruction-aligned autoencoder, such as a VAE [29] or recent variants [16, 71, 1], whose latents are optimized for pixel fidelity and stable decoding. But robotic world models are more than video generators, where planning and evaluations require predictions that preserve physical, spatial, and task dynamics. This motivates using the semantic spaces of self-supervised and vision-language encoders as latents for robot world modeling [11, 41, 22, 23, 4, 47, 61]. These spaces expose object layout and task structure more directly than pixel-trained autoencoders [53]. However, they are hard to use for diffusion due to their higher dimensionality yielding off-manifold latent generation with poor object structures [78]. RAE [79] makes them more tractable with a dimension-dependent noise-schedule shift and a wide DDT head [67], while S-VAE [78] learns a compact, KL-regularized latent space using an autoencoder as an adapter over the frozen semantic features.

Still, the effect of semantic latents on action-conditioned LDM for robotics remains open. DINO-WM [80] and V-JEPA 2-AC [4] show that pretrained feature spaces support planning, but they are not diffusion models: DINO-WM is an autoregressive feature-prediction world model, while V-JEPA 2-AC is a JEPA predictor [3]. RAE-NWM [76] shows that DINOv2 [41] spaces support diffusion-based navigation world modeling. Yet

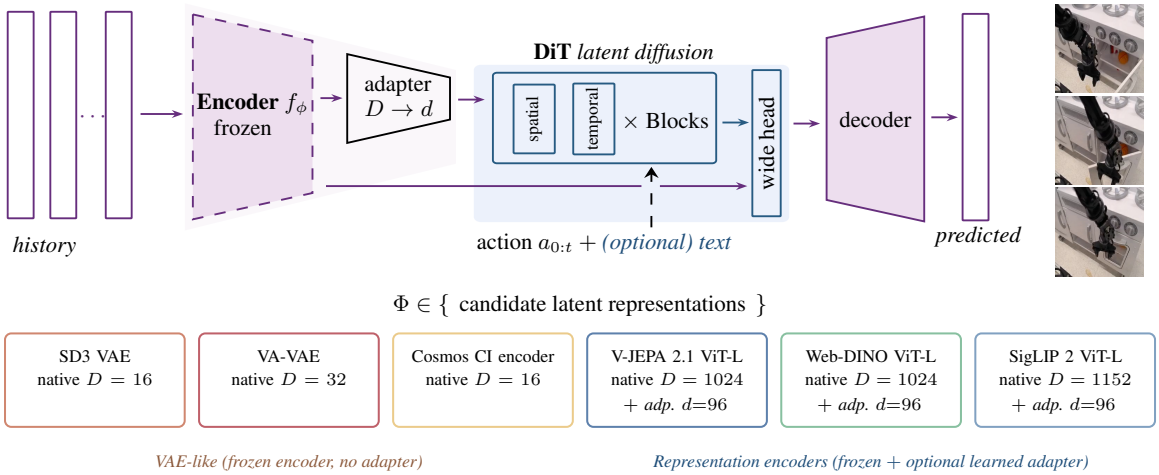



Figure 1: **Which latent space makes a better robotic world model?** For a latent diffusion model, we fix the Diffusion Transformer (DiT) transition model, action conditioning, and training data. We vary only the encoder $f_\phi$ defined latent interface: encoder, optional compression adapter, and the associated decoder path. This isolates how reconstruction-aligned and semantic representations affect action-faithful dynamics, generated rollouts, and downstream policy performance for robot control. We show the encoder families compared in the bottom panels.

navigation differs from contact-rich manipulation, where gripper motion, object state, geometry, and policy rollouts all matter. This leads to our question: **what effects does latent space choice have for LDM-based robotic world modeling?**

We answer this with a controlled evaluation study that varies only the representation space in which the transition model operates (see Fig. 1). For effective semantic space LDM training, we adapt RAE's wide-head and schedule-shift recipe [79] alongside the compact S-VAE adapter [78], and train on the Bridge V2 dataset [66] with the same DiT transition model [42] and action-conditioning scheme. We then propose an evaluation suite spanning three axes: visual fidelity, planning and downstream policy performance, and latent quality. Our findings show that semantic latents improve action recoverability, task-success classification, CEM planning, and policy-in-the-loop success, while reconstruction latents mainly retain photometric advantages. Our key contributions are three-fold:

1. Our primary contribution is the *evaluation* of representation spaces for latent diffusion world modeling. We do controlled analyses of how latent space choice affects not only visual generation, but also robotics tasks and robustness through our proposed three evaluation axes.
2. We propose an effective recipe for *training diffusion world models in high dimensional semantic spaces*, by leveraging the recent advances in semantic space diffusion and extending them to action-conditioned world modeling. We also study the effects of different design choices.
3. We show that semantic latent spaces are consistently more useful for policy evaluation and planning, even when reconstruction latents match or exceed them on low-level pixel fidelity, establishing that the best robotic world model latent space is the one that preserves action-relevant structure, not merely the one that reconstructs images the best.

## 2 Problem Formulation

We consider multi-task robot manipulation from partial observations. The offline dataset is $\mathcal{D} = \{(o_{0:T}, a_{0:T-1}, \ell, y)\}$, where $o_t \in \mathcal{O}$ is an RGB observation, $a_t \in \mathbb{R}^{d_a}$ is a continuous robot action, $\ell$ is an optional language instruction, and $y \in \{0, 1\}$ denotes episode success. Tasks vary in object configurations and instructions, but share a robot embodiment; we therefore view the data as samples from related partially observed Markov Decision Processes with shared dynamics and task-dependent goals. Because a single observation does not generally determine the next observation under an action, we condition on a fi-

nite visual-action history of length $H$ and model the action-conditioned predictive distribution over a rollout horizon $K$: $p(o_{t+1:t+K} \mid o_{t-H:t}, a_{t-H:t+K-1})$.

### 2.1 Latent Space World Models

Rather than predicting future frames directly in pixel space, latent world models learn predictive dynamics in a representation space. Each model consists of a frozen encoder, an optional frozen adapter, an action-conditioned transition model, and a decoder.

**Encoder and adapter.** A pretrained image encoder maps each observation to a spatial latent $z_t = f_\phi(o_t) \in \mathbb{R}^{N \times D}$, where $N = h \times w$ is the number of patches and $D$ is the encoder's native channel dimension. The encoder is frozen, so $f_\phi$ fixes the representation space in which dynamics are learned. For high-dimensional semantic representation encoders, we optionally use a frozen adapter $\alpha_\psi$ to obtain compact diffusion-friendly latents $\tilde{z}_t = \alpha_\psi(z_t) \in \mathbb{R}^{N \times d}$ [78]. For compressed reconstruction-aligned latent spaces, the adapter is the identity map.

**Transition model.** An action-conditioned DiT [42] predicts future latent trajectories: $\tilde{z}_{t+1:t+K} \sim p_\theta(\cdot \mid \tilde{z}_{t-H:t}, a_{t-H:t+K-1})$. Only the transition model is updated during world model training; the encoder, adapter, and decoder remain fixed. For semantic encoders without adapters, we add a lightweight wide DDT head [67], which adds few parameters but addresses the width bottleneck of DiT for high-dimensional latent spaces [79]. Otherwise, variants share the same transition backbone and differ only in representation and decoding path. Table 4 (Appx. B) shows that the DiT backbone with adapter does not incur an increase in parameter count or GFLOPs. Compute parity is explained in Appx. A.

**Decoder.** Predicted latents are mapped back to pixels as $\hat{o}_{t+1:t+K} = \mathrm{Dec}(\tilde{z}_{t+1:t+K})$. The decoder is needed for visual rollouts and pixel-level evaluation, but decoded image quality alone does not determine world model quality: a model may render plausible frames while missing action-relevant dynamics, or preserve control-relevant structure despite minor photometric errors.

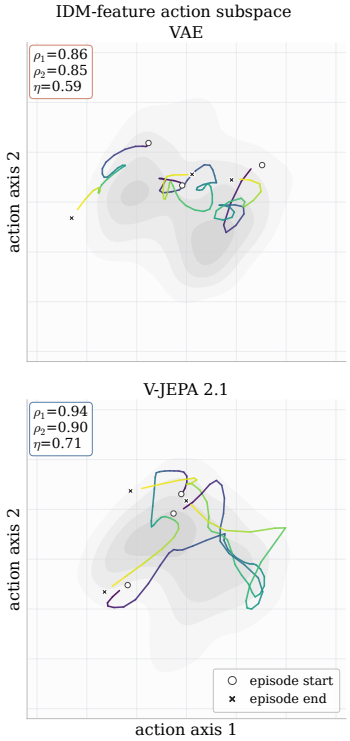



Figure 2: **Action trajectories induced by encoder spaces:** episode rollouts projected onto the top-2 canonical-correlation directions between IDM features and ground-truth actions. $(\rho_1, \rho_2)$ are the leading canonical correlations, $\eta$ summarizes the aggregate action alignment. Colored curves are episodes.

### 2.2 The Role of the Latent Space in Robotics

The encoder-defined latent space determines the state representation on which the transition model $p_\theta$ learns dynamics. In LDM, reconstruction-aligned latents $z_t^{\mathrm{pix}} = f_\phi^{\mathrm{PIX}}(o_t) \in \mathbb{R}^{N \times D_{\mathrm{pix}}}$ are commonly used because they preserve pixel-level information and provide reliable decoders [13]. For robotic world models, however, the relevant state is not only what an image looks like, but how it changes under actions and whether those changes preserve task progress, object state, contact, and geometry. This creates a multi-objective problem where useful latents should be action-controllable, task-informative, visually decodable, and useful for planning or policy evaluation.

As an initial diagnostic, we use inverse dynamics model (IDM) to probe whether an encoder makes action-relevant change explicit in latent space (see Appx. D.4 for details). Figure 2 shows that different encoders induce markedly different action-aligned trajectory geometries, suggesting that encoder choice changes which aspects of robot dynamics are easy for a transition model to learn. This motivates us to treat the latent space $f_\phi$ as the experimental variable, and evaluate its effect beyond visual fidelity and on axes spanning controllability, task semantics, and policy performance.

We thus compare reconstruction-aligned latents with semantic latents from pretrained vision foundation models [41, 4, 61], denoted as $z_t^{\mathrm{rep}} = f_\phi^{\mathrm{REP}}(o_t) \in \mathbb{R}^{N \times D_{\mathrm{rep}}}$. Since $D_{\mathrm{rep}}$ is typically large, we evaluate both

native features and compact adapter latents $\tilde{z}_t = \alpha_\psi(z_t^{\text{rep}})$. We train one world model per candidate in $\Phi = \{f_\phi^{(1)}, \ldots, f_\phi^{(m)}\}$ while fixing the data, history, action conditioning, optimizer, and transition backbone, so that each model learns a different latent transition $p_\theta^{(\phi)}(\tilde{z}_{t+1:t+K} \mid \tilde{z}_{t-H:t}, a_{t-H:t+K-1})$. The decoder differences are controlled through reconstruction gap metrics, latent-space metrics, and planning metrics.

# 3 Experiments

## 3.1 Dataset and Training

**Benchmark protocol.** We isolate the effect of the encoder-defined latent space by fixing the dataset, history length, action conditioning, transition architecture, optimizer, and training schedule, and varying only the encoder $f_\phi$, optional adapter $\alpha_\psi$, and decoder path. For each encoder–adapter pair, we train an LDM from scratch and evaluate the resulting world model for visual fidelity, representation quality, and downstream policy performance (see Appx. B).

**Dataset.** We train and evaluate on Bridge V2 [66], a real-robot manipulation dataset with ≈60K WidowX 250 demonstrations across 13 task families. Each episode includes RGB observations, 7 Degrees of Freedom (DoF) end-effector actions covering position, rotation, and gripper state, and a language instruction. For trajectory success classification, we use SOAR [81] which contains roughly 30.5K success/failure class episodes for WidowX 250 with a 1:2 class split.

**Encoder variants.** We compare two encoder families. reconstruction-aligned encoders $f_\phi^{\text{PIX}}$ include: Stable Diffusion 3 (SD3) VAE [16] with $D$=16, VA-VAE [71] with $D$=32, and Cosmos [1] with $D$=16; for these, $\alpha_\psi \equiv \mathbb{I}$. Semantics-aligned encoders $f_\phi^{\text{REP}}$ include: V-JEPA 2.1 [38] with $D$=1024, Web-DINO [18], adapted from DINOv2 [41], with $D$=1024, and SigLIP 2 [61] with $D$=1152. For semantic encoders, we evaluate both native latents and compact latents from a pretrained S-VAE adapter [78], which maps $D \to d$ with $d$=96.

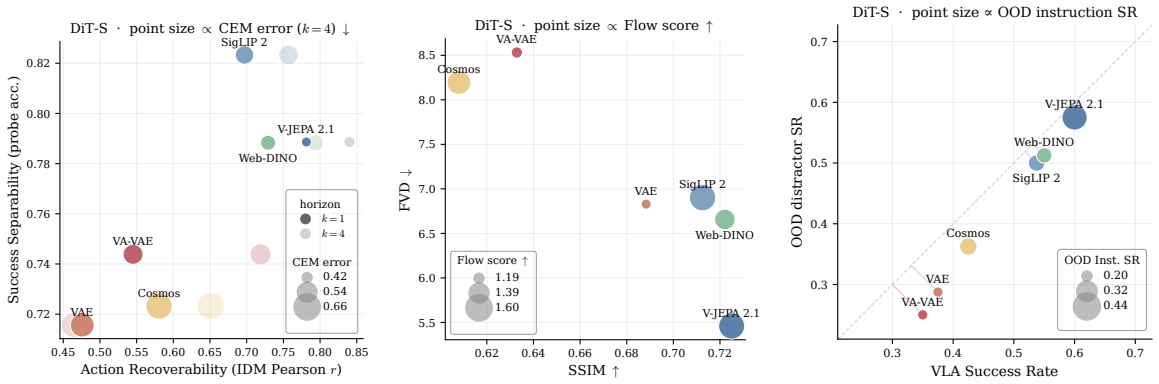
**Adapter, decoder, and transition model.** The S-VAE adapter [78] is pretrained to reconstruct frozen encoder features with a KL-regularized loss, and is paired with a lightweight pixel decoder. All transition models are DiTs trained on Bridge V2 [66] with flow matching [35]. Each DiT layer factorizes attention into a spatial block within each frame and a causal temporal block across frames. We sample every second frame, condition on $H$=2 history frames, and predict 8 future frames. We do not make use of language instruction conditioning while training the DiT. For all non-VAE encoders, we apply a dimension-dependent noise-schedule shift [16]. At inference, models roll out autoregressively one frame at a time using a 10-frame sliding context; VAE variants use their native pixel decoders, while semantic variants use the learned adapter decoder (see Appx. B for details).

## 3.2 Evaluation Metrics

To study how the choice of latent representation propagates through to downstream tasks, we propose an evaluation suite that segregates this effect across three axes. See Appx. C for details.

1. **Planning and downstream policy performance.** For robotics applications, a latent world model should enable planning, *i.e.*, searching for the optimal action sequence given a goal state [80, 4]. Evaluating planning helps separate the latent world modeling performance from the pixel decoder performance, which visual metrics conflate together. Given a real $k$-step transition, we use the cross-entropy method (CEM) [49] to recover the action sequence whose predicted latent best matches the target, and report CEM error at single-step $(k = 1)$ and multi-step $(k = 4)$ horizons.

   We also test whether the world model can serve as a policy-evaluation environment. We roll out OpenVLA-7B [28] inside each world model on 20 Bridge V2 test episodes with 8 trials per episode, and a subset of 10 of these were used for Out-Of-Distribution (OOD) evaluations. We use two Vision-Language Models (VLMs): InternVL 3.5 [68] and Qwen 3.6 [46], to judge the tasks' success. We report consensus success rate, Borda rank, and robustness under distractor-object and OOD-instruction perturbations. See Appx. C




(a) Latent space utility (b) Visual utility (c) Policy performance

Figure 3: **Latent space effect overview:** each point is a DiT-S world model trained by varying only the encoder and the associated decoder path. (a) **Upper-right is favorable.** Latent space metrics show that semantic encoders improve action recoverability, task-success separability, and action planning error (CEM) relative to reconstruction-aligned encoders. (b) **Lower-right is favorable.** Visual utility metrics show that pixel fidelity alone does not explain downstream performance: reconstruction-aligned spaces remain competitive on low-level image quality, while semantic spaces often improve video and motion quality. (c) **Upper-right is favorable.** Closed-loop evaluations show that semantic spaces generally yield higher VLA success and stronger robustness to OOD objects and instructions. Details about all metrics are in Sec. 3.2 and Appx. C.

for metrics definitions, Appx. A regarding fairness of VLM ratings, and Appx. C.4 & C.5 for exact details about OOD frame and OOD instruction generation, as well as details about tasks.

2. **Pixel fidelity and scene geometry.** Decoded rollouts must remain visually coherent to support visual policies. We report image/video metrics: FID, SSIM, LPIPS, FVD, temporal LPIPS, and point-track consistency, together with perceptual and geometric scores from WorldArena [51]. This family measures generation and motion quality, temporal consistency, and scene geometry.
3. **Latent representation quality.** Because the transition model operates in latent space, we directly probe whether generated latents preserve action and task-relevant structure. We train an inverse dynamics model (IDM) [57] on frozen encoder latents to recover action chunks for horizon $k \in \{1, 4\}$, and apply the IDM to world model latents to measure generation-induced degradation. We train a classifier on latent trajectories of SOAR [81], a language and success label annotated dataset of trajectories, to classify whether a trajectory was a success given the text instruction. We again measure the degradation in accuracy induced by evaluating on generated latents.

# 4 Findings

## 4.1 Does the choice of latent space affect planning and policy performance?

**Semantic latents offer better policy-in-the-loop performance.** Table 1 shows that encoder choice strongly affects downstream VLA policy rollouts at DiT-S. Reconstruction-aligned spaces perform worst: VAE and VA-VAE have the lowest consensus success rates and weakest Borda ranks, while semantic encoders improve policy success, interaction quality, and robustness. V-JEPA 2.1 and SigLIP 2 variants give the strongest DiT-S results. Semantic-family VLA SR and CEM outperform reconstruction-family under paired bootstrap over tasks as shown in our analysis in Appx. D.3.

**Native semantic spaces preserve action geometry for planning.** Representation aligned spaces have the lowest action-recovery errors across all DiT backbone sizes (Table 1, and Table 10 in Appx D). For example, at DiT-S V-JEPA 2.1 is best at $k{=}4$ and SigLIP 2 is best at $k{=}1$. Fig. 3c likewise shows semantic encoders closer to the upper-right diagonal in the VLA–OOD plane, while VAE-family models fall lower and suffer larger distractor-induced drops.

Table 1: **DiT-S policy and behavioral metrics.** **Best** and runner-up per column. In-distribution (ID) SR and Out-of-Distribution (OOD) SR are calculated on a subset of 10 episodes with InternVL 3.5. Consensus SR and Borda rank aggregate InternVL3.5-14B and Qwen3.6-27B rankings. Interaction quality measures the plausibility of robot-object contact. PCK coverage measures point tracking recall (Appx. C). Muted ± terms show one standard deviation error averaged over episodes.

| Encoder | VLA SR | | Interaction quality | | PCK | OOD robustness | | | CEM error | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Consensus SR ↑ | Borda rank ↓ | IQ score ↑ | Instr. follow ↑ | PCK coverage ↑ | ID SR ↑ | OOD SR distractor ↑ | OOD SR instruction ↑ | k=1 ↓ | k=4 ↓ |
| • VAE | 0.169 ±0.030 | 25 | 3.26 | 3.48 | 0.719 | 0.375 ±0.054 | 0.287 ±0.051 | 0.200 ±0.045 | 0.111 ±0.009 | 0.612 ±0.023 |
| • VA-VAE | 0.175 ±0.030 | 23 | 3.22 | 3.42 | 0.715 | 0.350 ±0.053 | 0.250 ±0.048 | 0.200 ±0.045 | 0.097 ±0.005 | 0.543 ±0.023 |
| • Cosmos | 0.244 ±0.034 | 16 | 3.32 | 3.51 | 0.707 | 0.425 ±0.055 | 0.362 ±0.054 | 0.275 ±0.050 | 0.112 ±0.009 | 0.661 ±0.033 |
| • V-JEPA 2.1 | 0.344 ±0.038 | **6** | 3.43 | 3.78 | **0.735** | 0.600 ±0.055 | 0.575 ±0.055 | **0.400** ±0.055 | 0.084 ±0.008 | **0.424** ±0.014 |
| • V-JEPA $2.1_{96}$ | **0.362** ±0.038 | 8 | **3.52** | **3.84** | 0.735 | 0.600 ±0.055 | 0.537 ±0.056 | 0.250 ±0.048 | 0.089 ±0.007 | 0.548 ±0.017 |
| • Web-DINO | 0.212 ±0.032 | 21 | 3.34 | 3.58 | 0.735 | 0.550 ±0.056 | 0.512 ±0.056 | 0.250 ±0.048 | 0.090 ±0.007 | 0.474 ±0.026 |
| • Web-DINO$_{96}$ | 0.300 ±0.036 | 11 | 3.44 | 3.77 | 0.732 | 0.600 ±0.055 | 0.512 ±0.056 | 0.275 ±0.050 | 0.090 ±0.007 | 0.531 ±0.025 |
| • SigLIP 2 | 0.325 ±0.037 | 9 | 3.43 | 3.58 | 0.730 | 0.537 ±0.056 | 0.500 ±0.056 | 0.263 ±0.049 | **0.082** ±0.006 | 0.523 ±0.030 |
| • SigLIP $2_{96}$ | 0.331 ±0.037 | 15 | 3.42 | 3.71 | 0.731 | **0.625** ±0.054 | **0.588** ±0.055 | 0.312 ±0.052 | 0.086 ±0.005 | 0.537 ±0.026 |

**Scaling narrows policy gaps but not action-centric gaps.** Appx. Table 10 shows that For DiT-L, the gaps in VLA success and OOD robustness for VAE and Cosmos narrow relative to semantic encoders. We attribute this to improved visual fidelity at larger model size, which benefits the VLA policy. However, both still lag on CEM action recovery, which depends directly on latent transition structure rather than rendered visual quality; at DiT-L, VAE and Cosmos have larger $k$=1 CEM errors than all semantic encoders. They also lag on IDM $r$ and classifier accuracy (Table 13 and 14).

### 4.2 Does the latent space affect action recoverability and preservation of task semantics?

**Semantic latents make action-relevant changes more recoverable.** Table 2 shows that semantic encoders retain substantially more action information than reconstruction-aligned ones. On encoder latents, V-JEPA 2.1 and Web-DINO achieve the strongest IDM Pearson $r$ across both horizons, and this advantage largely persists after world model (WM) generation. The trends also hold with DiT scaling (Tables 13 and 14 in Appx. D.4).

Table 2: **IDM Pearson** $r$ (horizons $k \in \{1, 4\}$) and Success classifier for DiT-S, reported on encoder (Enc.) and world model (WM) latents.

| Encoder | Pearson $r$ | | | | Classifier Acc. | |
|---|---|---|---|---|---|---|
| | Enc.↑ | | WM↑ | | Whole-video | |
| | $k$=1 | $k$=4 | $k$=1 | $k$=4 | Enc.↑ | WM↑ |
| • VAE | 0.507 | 0.478 | 0.476 | 0.464 | 0.835 | 0.716 |
| • VA-VAE | 0.549 | 0.744 | 0.545 | 0.719 | 0.868 | 0.744 |
| • Cosmos | 0.626 | 0.673 | 0.581 | 0.651 | 0.851 | 0.723 |
| • V-JEPA 2.1 | **0.829** | **0.865** | **0.781** | **0.840** | 0.905 | 0.789 |
| • Web-DINO | 0.820 | 0.845 | 0.729 | 0.794 | **0.906** | 0.788 |
| • SigLIP 2 | 0.772 | 0.793 | 0.697 | 0.757 | 0.903 | **0.823** |

**Semantic latents better preserve task-success information.** From Table 2, we also see that success classifiers trained on frozen encoder latents achieve higher accuracy for semantic encoders, and their performance degrades less when evaluated on generated WM latents, with SigLIP 2 having best WM latent accuracy. This indicates that semantic spaces not only encode local action effects, but also retain higher-level task progress signals useful for policy evaluation.

### 4.3 How does the latent space affect visual fidelity?

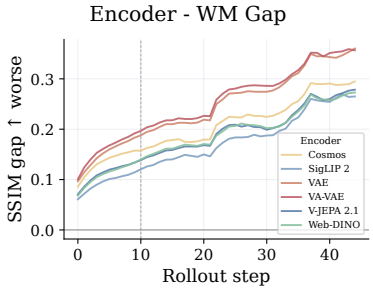



Figure 4: **SSIM gap** over steps.

**Semantic latent spaces remain visually competitive.** Table 3 shows that the policy gains from semantic encoders do not come at the cost of decoded visual quality. At DiT-S scale, these encoders dominate most perceptual, structural, and video-level metrics, particularly when used with adapters $d_{96}$: SigLIP $2_{96}$ gives the best SSIM, V-JEPA $2.1_{96}$ gives the best FVD, and Web-DINO variants are strongest on JEPA similarity, subject consistency, depth error, and temporal LPIPS.

VAE-style spaces remain competitive on image quality, and qualitatively tend to preserve sharper local appearance details, but they lag behind

semantic spaces on global structure and temporal generation quality. Figures 4 and 9 (Appx. D) show semantic space models have lower gap for pixel reconstruction, particularly while extrapolating beyond the 10-frame horizon length seen during training.

**Large DiTs help recover much of the visual advantage of reconstruction latents.** Increasing transition model capacity benefits reconstruction latents the most. For DiT-L, VAE becomes highly competitive, achieving the best FID, image quality, aesthetic quality, JEPA similarity, depth error, dynamic degree, and FVD, while also ranking second on LPIPS and flow score. Here, semantic encoders still remain strong: V-JEPA $2.1_{96}$ gives the best SSIM and LPIPS, and SigLIP $2_{96}$ remains competitive on structure and temporal metrics, but their gains from scaling are less uniform. Overall, visual fidelity alone does not explain the downstream policy advantages observed in Sec. 4.1.

### 4.4 Does scaling along input views and model size help?

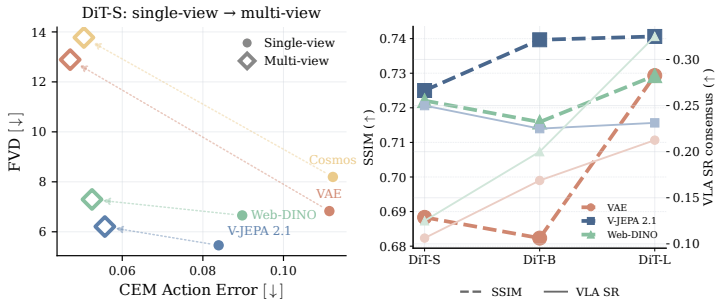



Figure 5: **Scaling** camera views (left) and DiT sizes (right).

**Multi-view training improves action recovery but can hurt video quality under limited data.** We take the trained DiT-S models and finetune them for 20 epochs on the BridgeV2 episodes that contain three camera views. Fig. 5 (left) shows that while this does lead to superior CEM action prediction, it also degrades generation quality, possibly due to smaller number of training episodes. However, the semantic encoders are more robust to this degradation. **Model scaling improves both visual quality and policy success, with larger gains for reconstruction latents:** in Fig. 5 (right), we see that both generation (SSIM) and policy performance (VLA-SR) generally scale with the DiT size. Here, VAE scales notably well on visual metrics and approaches semantic encoders, which already perform strongly at DiT-S.

### 4.5 Do reconstruction-aligned and semantic encoders fail differently?

**The main failure modes differ: reconstruction latents hallucinate task semantics, while semantic latents miss geometry and contact.** Our qualitative rollouts in Appx. fig. 13 show that all encoder families share a common failure mode where static scene elements are faithfully preserved while manipulation-relevant details hallucinate. Beyond this universal pattern, encoder families show distinct hallucinations. Reconstruction encoders tend to fail at the object-semantic level: VAE and Cosmos hallucinate the white basket and green towel respectively in Fig. 7 producing coherent looking but task-incorrect states, and under OOD instructions (Appx. Fig. 17), both maintain the prior action pattern rather than updating to the new goal. Semantic encoders preserve task-level intent at the cost of geometric precision (e.g., VJEPA2.1 under-opens the drawer in Appx. Fig. 13). We find the latter to better capture semantic distinctions even under instruction shift (e.g., the fold-unfold task in Appx. Fig. 15).

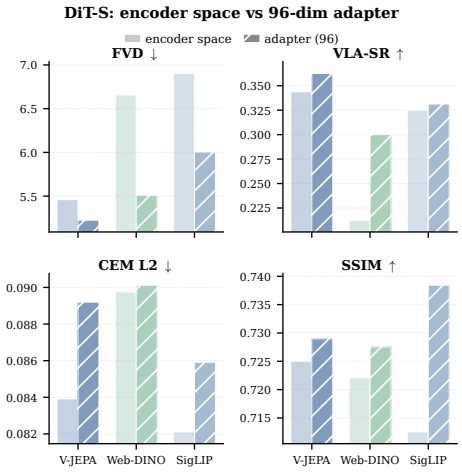



Figure 6: **Adapter** ablation results.

### 4.6 Do compressed adapter latents aid semantic encoders further for world modeling?

**Adapters improve diffusion ease but can distort control geometry.** Fig. 6, Table 1, and Table 3 show that the compressed space $d_{96}$ of adapters helps the latent diffusion model, as also observed by Zhang et al.

[78] and Bai et al. [5]. This leads to generally stronger performance than the native variants on most metrics except latent CEM action error, OOD robustness, and PCK coverage. These findings hint towards the adapter compressing the latent space in a way that is useful for high-level task completion such as diffusion denoising but hurtful for fine-grained tasks like trajectory optimization, where precise action information is needed.

### 4.7 Do high-dimensional semantic latents and adapter add computational overhead?

**High-dimensional semantic latents do not substantially increase DiT compute in our setup.** The DiT always receives the same number of tokens per frame $N{=}256$, hence larger channel dimensions only affect the input/output projections (see Appx. B.2 for discussion). The main compute differences instead come from the frozen encoder and decoder architectures. In particular, ViT-based semantic encoders paired with the adapter pixel decoder remain competitive in total GFLOPs, while native high-dimensional semantic spaces require only a lightweight wide DDT head [67]. We report parameter counts and GFLOPs split by encoder, adapter, DiT, and decoder in Appx. Table 4.

**Key Empirical Takeaways**

- **Visual fidelity does not always imply downstream performance.** Reconstruction latents can match or exceed semantic latents on pixel-level metrics, especially at larger DiT scale, yet lag on action recovery, task-success probes, CEM planning, and policy-in-the-loop evaluation.
- **Semantic latents scale better with multiple views.**
  Under limited data, adding multiple views improves planning but can hurt visual rollouts; semantic encoders retain the action recoverability benefit with substantially less degradation than reconstruction latents.
- **Adapters trade control geometry for diffusion ease.** Adapters ease diffusion and decoding, but can distort fine-grained action geometry compared with native semantic features.
- **World models in semantic spaces lower reconstruction and generation ceiling gap.** Training decoders with the same budget for semantic world models is more effective.
- **High-dimensional semantic latents are practical in DiTs.** With a fixed patch-token count, semantic width adds little to the transition-model cost.

## 5 A Recipe for Semantic Latent Diffusion Robotics World Modeling

Our findings suggest a practical recipe for building robotic latent diffusion world models. **Do not begin by optimizing for visual realism alone**. Instead, choose a latent space that makes **action and task progress** explicit, make that space easy for diffusion to model, and evaluate the resulting world model with control- and policy-based metrics. Visual realism can often be improved through better decoder training, but transition quality and latent fidelity remain important. Use robot demonstration datasets with preferably **multi-view trajectories** and, when available, success/failure labels to unlock diverse evaluations. **Choose pretrained semantic encoders** as the default latent state space, since they preserve action geometry and task progress better than reconstruction latents. **Pair them with adapter compression** when decoded rollout quality or VLA-in-the-loop evaluation matters. For transition model, a robust default for high-dimensional semantic spaces is: a **spatial-temporal DiT** with causal temporal blocks, a shallow-wide DDT head [67], and a dimension-aware noise shifting [79]. The spatial blocks stay non-causal since per-frame patches are denoised jointly. For training, diffusion forcing [12] can be used for autoregressive next-frame rollout. Finally, evaluate world models on **multiple axes** covering both visual, latent, and downstream task performance.

## 6 Related work

**Robotic world models** can be seen to span three related objectives. One line treats world models as policy-evaluation environments: WorldGym [44] and WorldEval [34] roll out policies in learned video models; [62] studies how pretraining, data diversity, and failure modes affect evaluation. A second line adapts pretrained generators into interactive simulators: UniSim [70] learns interactive real-world simulators from broad data;

Table 3: **Visual realism quality** for DiT-S and L. **Best** and runner-up within each size group.

| Encoder | Visual quality | | | | | | Content consistency | | Motion quality | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SSIM↑ | LPIPS↓ | FID↓ | Image quality↑ | Aesthetic quality↑ | JEPA sim.↑ | Subject consist.↑ | Depth AbsRel↓ | Dyn. degree↑ | Flow score↑ | FVD↓ | t-LPIPS↓ |
| DiT-S | | | | | | | | | | | | |
| VAE | 0.688 | 0.218 | 17.428 | **0.592** | 0.467 | 0.871 | 0.810 | 0.390 | 0.767 | 1.186 | 6.829 | 0.0264 |
| VA-VAE | 0.633 | 0.226 | 15.488 | 0.585 | 0.464 | 0.783 | 0.817 | 0.455 | 0.765 | 1.204 | 8.531 | 0.0253 |
| Cosmos | 0.608 | 0.245 | 16.947 | 0.558 | 0.463 | 0.517 | 0.793 | 0.638 | 0.813 | 1.511 | 8.195 | 0.0223 |
| V-JEPA 2.1 | 0.725 | **0.176** | 6.771 | 0.578 | 0.473 | 0.929 | 0.841 | 0.404 | 0.832 | 1.587 | 5.459 | 0.0197 |
| V-JEPA $2.1_{96}$ | 0.729 | 0.179 | 6.302 | 0.579 | **0.474** | 0.928 | 0.841 | 0.363 | **0.843** | **1.653** | **5.224** | 0.0212 |
| Web-DINO | 0.722 | 0.199 | 7.626 | 0.576 | 0.472 | 0.938 | **0.849** | **0.350** | 0.794 | 1.408 | 6.656 | 0.0234 |
| Web-$DINO_{96}$ | 0.728 | 0.181 | **5.998** | 0.574 | 0.473 | **0.944** | 0.841 | 0.375 | 0.835 | 1.634 | 5.510 | **0.0195** |
| SigLIP 2 | 0.713 | 0.205 | 7.858 | 0.566 | 0.471 | 0.931 | 0.839 | 0.394 | 0.827 | 1.602 | 6.902 | 0.0228 |
| SigLIP $2_{96}$ | **0.738** | 0.179 | 6.881 | 0.573 | 0.472 | 0.938 | 0.843 | 0.372 | 0.827 | 1.547 | 6.005 | 0.0223 |
| DiT-L | | | | | | | | | | | | |
| VAE | 0.729 | 0.168 | **5.351** | **0.598** | **0.475** | **0.980** | 0.827 | **0.281** | **0.844** | 1.635 | **3.495** | 0.0202 |
| Cosmos | 0.657 | 0.186 | 9.234 | 0.578 | 0.469 | 0.760 | 0.817 | 0.465 | 0.843 | **1.650** | 6.536 | 0.0199 |
| V-JEPA 2.1 | 0.741 | 0.172 | 6.944 | 0.578 | 0.474 | 0.926 | 0.844 | 0.330 | 0.832 | 1.573 | 5.371 | 0.0195 |
| V-JEPA $2.1_{96}$ | **0.743** | **0.165** | 6.186 | 0.581 | 0.474 | 0.929 | 0.842 | 0.346 | 0.831 | 1.558 | 5.223 | 0.0201 |
| Web-DINO | 0.729 | 0.192 | 6.918 | 0.573 | 0.472 | 0.945 | 0.847 | 0.343 | 0.823 | 1.557 | 6.014 | 0.0219 |
| Web-$DINO_{96}$ | 0.741 | 0.189 | 14.259 | 0.578 | 0.466 | 0.709 | **0.852** | 0.352 | 0.833 | 1.568 | 13.107 | **0.0189** |
| SigLIP 2 | 0.730 | 0.188 | 7.574 | 0.569 | 0.472 | 0.937 | 0.845 | 0.344 | 0.822 | 1.562 | 6.688 | 0.0207 |
| SigLIP $2_{96}$ | 0.743 | 0.171 | 6.740 | 0.573 | 0.472 | 0.937 | 0.844 | 0.326 | 0.830 | 1.580 | 5.780 | 0.0193 |

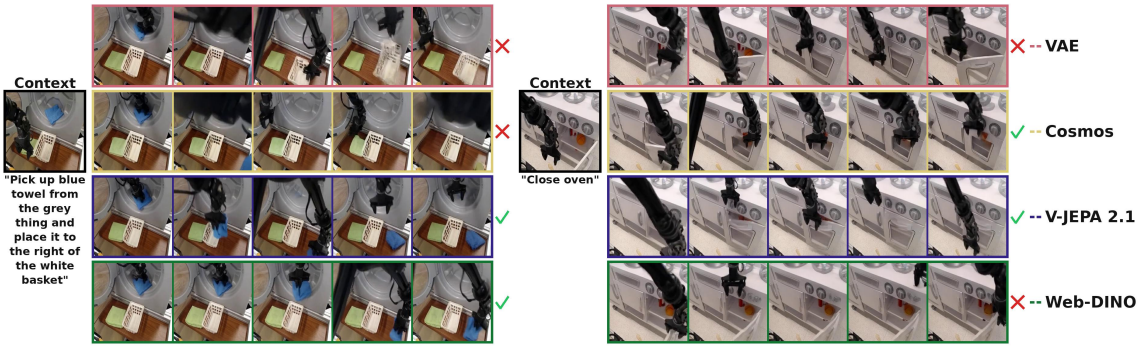



Figure 7: **Open-VLA success rate comparison on two random episodes:** four frames are sampled at even intervals. ✓ and ✗ show trajectories marked as success and failure by InternVL 3.5 VLM.

Vid2World [25] causalizes video diffusion with action guidance; Ctrl-World [19] studies multi-view, long-horizon, policy-in-the-loop manipulation. A third line moves prediction and planning into semantic feature space: DINO-WM [80], DINO-world [8], and V-JEPA 2-AC [4] show that pretrained representations can support latent space forecasting and zero-shot or few-shot planning. These works establish the utility of both video generation and semantic representations, but do not isolate the encoder-defined latent space within a unified action-conditioned framework.

**World model evaluation** has moved beyond rollout plausibility and policy ranking toward physics, semantics, and embodied utility [33, 38]. RBench [31] measures task correctness and structural realism. WorldModelBench [32] highlights instruction-following and physics-adherence failures missed by generic video metrics. EWMBench [73] evaluates scene consistency, motion correctness, and semantic alignment. World-in-World [74] prioritizes closed-loop task success, WoW-World-Eval [17] adds inverse-dynamics-based action plausibility, and WorldArena [51] exposes the gap between perceptual quality and downstream functionality. These benchmarks evaluate world models at system-level while we seek to evaluate them at model-level. See Appx. B.1 for a review of LDM.

## 7 Future Work and Limitations

Our study isolates the effect of encoder-defined latent spaces within a controlled action-conditioned LDM protocol. The conclusions are therefore scoped to the Bridge V2 manipulation setting and a shared robot embodiment. Evaluating broader embodiments, domains, and data regimes is an important next step. Our policy-in-the-loop experiments also focus on evaluating a fixed VLA policy inside generated rollouts, while

policy improvement and sim-to-real transfer would test a complementary use of the same world models. Lastly, our evaluation partially relies on VLM-based success judgments, which may introduce evaluator bias. We reduce this dependence by aggregating multiple VLMs and pairing them with non-VLM diagnostics, including CEM planning, inverse dynamics, latent success classification, and visual/geometric metrics.

## 8 Conclusion

Our study shows that the encoder-defined latent space is a central design choice for action-conditioned latent diffusion world models in robotics. Across visual, latent, planning, and policy-in-the-loop evaluations, semantic representation spaces such as that of V-JEPA 2.1, Web-DINO, and SigLIP 2 generally provide stronger action recoverability, task-success classification accuracy, robustness, and downstream policy performance than reconstruction-aligned VAE-style latents, even when the latter remains competitive or superior on low-level photometric metrics. These results support the view that robotic world models should not be selected solely by visual realism, but by whether their latent dynamics preserve action-relevant structure and policy evaluation accuracy.

## Acknowledgments and Disclosure of Funding

Nilaksh is partly supported by a grant (`https://doi.org/10.69777/2009238`) from the Fonds de recherche du Québec (FRQNT). Saurav Jha is supported by the IVADO postdoctoral fellowship and the Canada First Research Excellence Fund. Sarath Chandar is supported by the Canada CIFAR AI Chairs program, the Canada Research Chair in Lifelong Machine Learning, and the NSERC Discovery Grant. This research was enabled in part by compute resources provided by Mila (`mila.quebec`) and the Digital Research Alliance of Canada (`www.alliancecan.ca`).

## References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[2] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, et al. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM international conference on architectural support for programming languages and operating systems, volume 2*, pages 929–947, 2024.

[3] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15619–15629, 2023.

[4] 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, June 2025.

[5] Jianhong Bai, Xiaoshi Wu, Xintao Wang, Xiao Fu, Yuanxing Zhang, Qinghe Wang, Xiaoyu Shi, Menghan Xia, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Kun Gai. SemanticGen: Video Generation in Semantic Space, December 2025.

[6] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang,

Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[7] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[8] 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.

[9] Homanga Bharadhwaj, Kevin Xie, and Florian Shkurti. Model-predictive control via cross-entropy and gradient-based optimization. In *Learning for Dynamics and Control*, pages 277–286. PMLR, 2020.

[10] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. 2024. *URL https://openai. com/research/video-generation-models-as-world-simulators*, 3(1):3, 2024.

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[12] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

[13] Rewon Child. Very deep {vae}s generalize autoregressive models and can outperform them on images. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=RLRXCV6DbEJ`.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

[15] Tom Erez, Yuval Tassa, and Emanuel Todorov. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 4397–4404. IEEE, 2015.

[16] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[17] Chun-Kai Fan, Xiaowei Chi, Xiaozhu Ju, Hao Li, Yong Bao, Yu-Kai Wang, Lizhang Chen, Zhiyuan Jiang, Kuangzhi Ge, Ying Li, et al. Wow, wo, val! a comprehensive embodied world model evaluation turing test. *arXiv preprint arXiv:2601.04137*, 2026.

[18] David Fan, Shengbang Tong, Jiachen Zhu, Koustuv Sinha, Zhuang Liu, Xinlei Chen, Michael Rabbat, Nicolas Ballas, Yann LeCun, Amir Bar, et al. Scaling language-free visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 370–382, 2025.

[19] Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

[20] David Ha and Jürgen Schmidhuber. World models. *eprint arXiv: 1803.10122*, 2018.

[21] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[23] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[24] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[25] Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2world: Crafting video diffusion models to interactive world models. *arXiv preprint arXiv:2505.14357*, 2025.

[26] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *Proc. ECCV*, 2024.

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

[28] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, pages 2679–2713. PMLR, 2025.

[29] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[31] Chunyi Li, Jianbo Zhang, Zicheng Zhang, Haoning Wu, Yuan Tian, Wei Sun, Guo Lu, Xiaohong Liu, Xiongkuo Min, Weisi Lin, and Guangtao Zhai. R-bench: Are your large multimodal model robust to real-world corruptions?, 2024.

[32] Dacheng Li, Yunhao Fang, Yukang Chen, Shuo Yang, Shiyi Cao, Justin Wong, Michael Luo, Xiaolong Wang, Hongxu Yin, Joseph E. Gonzalez, Ion Stoica, Song Han, and Yao Lu. Worldmodelbench: Judging video generation models as world models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL `https://openreview.net/forum?id=a3hafrDzuA`.

[33] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. In *Conference on Robot Learning*, pages 3705–3728. PMLR, 2025.

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

[35] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[37] Haoyu Lu, Guoxing Yang, Nanyi Fei, Yuqi Huo, Zhiwu Lu, Ping Luo, and Mingyu Ding. VDT: General-purpose video diffusion transformers via mask modeling. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Un0rgm9f04.

[38] 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.

[39] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

[40] Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *International Conference on Learning Representations (ICLR)*, 2026.

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

[42] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[43] Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient cross-entropy method for real-time planning. In *Conference on Robot Learning*, pages 1049–1065. PMLR, 2021.

[44] Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. Worldgym: World model as an environment for policy evaluation. *arXiv preprint arXiv:2506.00613*, 2025.

[45] Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

[46] Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026. URL https://qwen.ai/blog?id=qwen3.6-27b.

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[48] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[49] Reuven Y Rubinstein and Dirk P Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*, volume 133. Springer, 2004.

[50] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

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

[52] Ansh Kumar Sharma, Yixiang Sun, Ninghao Lu, Yunzhe Zhang, Jiarao Liu, and Sherry Yang. World-gymnast: Training robots with reinforcement learning in a world model. *arXiv preprint arXiv:2602.02454*, 2026.

[53] Minglei Shi, Haolin Wang, Wenzhao Zheng, Ziyang Yuan, Xiaoshi Wu, Xintao Wang, Pengfei Wan, Jie Zhou, and Jiwen Lu. Latent diffusion model without variational autoencoder. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=kdpeJNbFyf`.

[54] Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=2hEDcA7xy4`.

[55] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

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

[57] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=meRCKuUpmc`.

[58] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[59] Shengbang Tong, David Fan, John Nguyen, Ellis Brown, Gaoyue Zhou, Shengyi Qian, Boyang Zheng, Théophane Vallaeys, Junlin Han, Rob Fergus, et al. Beyond language modeling: An exploration of multimodal pretraining. *arXiv preprint arXiv:2603.03276*, 2026.

[60] Shengbang Tong, Boyang Zheng, Ziteng Wang, Bingda Tang, Nanye Ma, Ellis Brown, Jihan Yang, Rob Fergus, Yann LeCun, and Saining Xie. Scaling text-to-image diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2601.16208*, 2026.

[61] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[62] Wei-Cheng Tseng, Jinwei Gu, Qinsheng Zhang, Hanzi Mao, Ming-Yu Liu, Florian Shkurti, and Lin Yen-Chen. Scalable policy evaluation with video world models. *arXiv preprint arXiv:2511.11520*, 2025.

[63] 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.

[64] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302, 2021.

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[66] Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 1723–1736. PMLR, 06–09 Nov 2023. URL `https://proceedings.mlr.press/v229/walke23a.html`.

[67] Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. *arXiv preprint arXiv:2504.05741*, 2025.

[68] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[69] 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.

[70] Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *NeurIPS Workshop on Generalization in Planning*, 2023.

[71] 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*, 2025.

[72] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *Advances in Neural Information Processing Systems*, 37:128940–128966, 2024.

[73] Hu Yue, Siyuan Huang, Yue Liao, Shengcong Chen, Pengfei Zhou, Liliang Chen, Maoqing Yao, and Guanghui Ren. Ewmbench: Evaluating scene, motion, and semantic quality in embodied world models. *arXiv preprint arXiv:2505.09694*, 2025.

[74] Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal M Patel, Paul Pu Liang, et al. World-in-world: World models in a closed-loop world. *arXiv preprint arXiv:2510.18135*, 2025.

[75] Jiahui Zhang, Ze Huang, Chun Gu, Zipei Ma, and Li Zhang. Reinforcing action policies by prophesying. *arXiv preprint arXiv:2511.20633*, 2025.

[76] Mingkun Zhang, Wangtian Shen, Fan Zhang, Haijian Qin, Zihao Pei, and Ziyang Meng. Rae-nwm: Navigation world model in dense visual representation space. *arXiv preprint arXiv:2603.09241*, 2026.

[77] 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*, pages 586–595, 2018.

[78] Shilong Zhang, He Zhang, Zhifei Zhang, Chongjian Ge, Shuchen Xue, Shaoteng Liu, Mengwei Ren, Soo Ye Kim, Yuqian Zhou, Qing Liu, et al. Both semantics and reconstruction matter: Making representation encoders ready for text-to-image generation and editing. *arXiv preprint arXiv:2512.17909*, 2025.

[79] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

[80] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World Models on Pre-trained Visual Features enable Zero-shot Planning. In *Forty-Second International Conference on Machine Learning*, June 2025.

[81] Zhiyuan Zhou, Pranav Atreya, Abraham Lee, Homer Walke, Oier Mees, and Sergey Levine. Autonomous improvement of instruction following skills via foundation models. *arXiv preprint arXiv:407.20635*, 2024.

[82] Fangqi Zhu, Zhengyang Yan, Zicong Hong, Quanxin Shou, Xiao Ma, and Song Guo. Wmpo: World model-based policy optimization for vision-language-action models. *arXiv preprint arXiv:2511.09515*, 2025.

# A Frequently Asked Questions (FAQs)

1. **What are the parameter counts and GFLOPs of the full diffusion pipelines for each of the encoder families? How is the parameter/compute parity ensured with adapters and wide heads?**

   We show in Table 4 in Appx. B the summary of the parameter counts and compute required for inference of all semantic spaces, with and without adapters.

   Parameter and compute parity are ensured by keeping the same DiT backbone across all rows and giving every model the same 256 tokens per frame. For adapter-based semantic encoders, the S-VAE adapter compresses high-dimensional features to 96 channels, making the DiT almost identical to the VAE-latent case. For native semantic latents, only the shallow input/output projection or wide head changes, so the *extra parameters do not increase DiT depth and add little compute*. Thus, the comparison is not driven by a larger diffusion model, it isolates the effect of using richer semantic representation spaces, which remain competitive in compute while providing stronger task-relevant structure.

2. **How sensitive are the policy-in-the-loop results to the choice of VLM judges? Are inter-judge agreements available?**

   The policy-in-the-loop results do show sensitivity to the VLM judge, particularly on harder tasks: agreement is high on simple Level 1 tasks, while Level 2–4 tasks involve finer spatial, contact, deformable-object, and stacking judgments that naturally induce more judge variation; see Table 11 for detailed results. We therefore rate trajectories with three VLMs and select the two most correlated judges, InternVL3.5-14B and Qwen3.6-27B, based on inter-judge Cohen's $\kappa$ agreement (Fig. 8). To further reduce single-judge dependence, Table 1 reports both consensus success rates with variance and Borda ranks, which are less sensitive to absolute score calibration. Finally, our conclusions do not rely only on VLM ratings: we also report task-instruction-conditioned success-classifier metrics on generated latents in Table 2, providing an independent task-conditioned signal that supports the same trends.

3. **Why was CEM chosen for latent space planning instead of gradient based planners or differentiable MPC?**

   We use CEM because latent-space planning involves non-convex objectives and noisy gradients. As a derivative-free optimizer, CEM is robust to black-box dynamics and compounding errors [49]. Its stochastic search avoids local minima better than gradient-based or differentiable Model Predictive Control (MPC), motivating its use in PlaNet [21] and CEM-MPC [43]. While gradient planners are faster, they are sensitive to model inaccuracies and gradient instability [9]. Consequently, CEM provides a conservative, reliable baseline for evaluating world-model quality.

4. **Is there evaluation on another manipulation dataset or embodiment (e.g., ALOHA, Franka) to test generalization? What are the expected transfer and potential pitfalls?**

   Evaluation on additional embodiments is an important direction but outside the scope of this study, whose controlled comparison is centered on BridgeV2; we did, however, use SOAR data for training the success classifier. We expect the main conclusion, that semantic latents are more policy-relevant than purely reconstruction-aligned latents, to transfer most directly when object-centric semantics and action-conditioned contact dynamics remain comparable. Cross-embodiment evaluation on ALOHA-style bimanual manipulation, Franka setups, or broader simulators such as RoboCasa [39, 40] would introduce new challenges: different camera viewpoints, action spaces, gripper morphology, control frequencies, embodiment-specific failure modes, and sim-to-real gaps. These factors may require embodiment-specific action tokenization, calibration, or classifier re-training, making such benchmarks an excellent test of whether semantic latent world models generalize beyond a single robot-data distribution. We also mention this as a potential future work avenue in Section 7.

5. **What is the benefit of diffusion models over non-diffusion world models that use semantic features for manipulation like DINO-WM and V-JEPA 2 AC?**

   DINO-WM and V-JEPA 2-AC provide compelling evidence that pretrained semantic features are useful for robotic prediction and planning, and we view them as complementary to our study rather than direct competitors. Our central research question is specifically how the choice of latent space affects *diffusion-based* action-conditioned world modeling, so comparing against non-diffusion architectures would conflate representation choice with model-family differences. Diffusion models are also a natural testbed for

this question because they model a distribution over future sequences and can denoise an entire prediction horizon jointly, which may better capture multimodal futures and reduce the compounding errors associated with purely autoregressive one-step regression rollouts, although this is a mitigation rather than a guarantee. Thus, our experiments are intentionally scoped to isolate the effect of semantic versus reconstruction latents within a fixed LDM framework; broader comparisons to non-diffusion semantic world models are important future work.

6. **How were the learning rates and other hyperparameters chosen for different encoder latent spaces?**

   We used the same optimizer and learning-rate recipe for all world models, rather than tuning separately for each latent space. Specifically, all DiTs were trained with AdamW, learning rate $10^{-4}$, betas $(0.9, 0.99)$, weight decay $2 \times 10^{-3}$, gradient clipping, EMA, linear warmup, and cosine decay. Our goal is to isolate the effect of the encoder-defined latent space, and per-encoder hyperparameter tuning would confound the comparison by giving different latent spaces different optimization budgets. For each model-size group, runs were trained under the same schedule and until losses had plateaued. Since each DiT-S run costs roughly 6–7 hours on 4 H100s, each DiT-L run about 34 hours, and adapter/pixel-decoder training about 55 hours, exhaustive sweeps over learning rate, weight decay, warmup, batch size, EMA, and noise schedule for every encoder would be prohibitively expensive. We therefore use a fixed standard recipe and report all models under the same optimization protocol.

# B Architecture and Training Details

Table 4: **Architecture size and compute.** Adapter-based semantic encoders are marked with $_{96}$ and use the S-VAE adapter with $d$=96. Native semantic rows do not use adapter in the DiT and use a shallow-wide DDT head. All DiT parameter counts are for DiT-L. Note that the extra DiT parameters are due to the shallow-wide head, which does not contribute much to the depth of the DiT. For DiTs using high-dimensional latents of V-JEPA 2.1, Web-DINO, and SigLIP, decoding uses the adapter's pixel decoder as the surrogate.

| | | | Params (M) | | | | GFLOPs/frame | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Encoder | DiT latent | Adapter | Enc. | Adapt. | DiT | Dec. | Enc. | Adapt. | DiT | Dec. | Total |
| SD-VAE | 16 | – | 34.3 | – | 910.1 | 49.5 | 270.8 | – | 316.5 | 620.2 | 1207.5 |
| Cosmos-CI16x16 | 16 | – | 33.5 | – | 910.0 | 48.0 | 47.6 | – | 316.5 | 101.7 | 465.9 |
| VA-VAE | 32 | – | 28.4 | – | 910.0 | 41.4 | 137.9 | – | 316.5 | 252.2 | 706.6 |
| V-JEPA $2.1_{96}$ | 96 | S-VAE | 304.7 | 38.1 | 910.1 | 177.0 | 154.7 | 10.6 | 316.5 | 428.3 | 910.1 |
| V-JEPA 2.1 | 1024 | – | 304.7 | – | 921.5 | 177.0 | 154.7 | – | 318.7 | 428.3 | 901.7 |
| Web-DINO$_{96}$ | 96 | S-VAE | 303.7 | 38.1 | 910.1 | 177.0 | 155.8 | 10.6 | 316.5 | 428.3 | 911.2 |
| Web-DINO | 1024 | – | 303.7 | – | 921.5 | 177.0 | 155.8 | – | 318.7 | 428.3 | 902.8 |
| SigLIP $2_{96}$ | 96 | S-VAE | 427.7 | 46.4 | 910.1 | 177.0 | 211.9 | 12.8 | 316.5 | 428.3 | 969.5 |
| SigLIP 2 | 1152 | – | 427.7 | – | 921.9 | 177.0 | 211.9 | – | 318.8 | 428.3 | 959.0 |

## B.1 Latent Diffusion Modeling (LDM)

LDM learns to denoise in compact reconstruction-aligned autoencoder spaces such as that of VAEs [29]. Recent VAE variants include: Stable Diffusion 3 [16] adapting autoencoding to rectified flow models, VA-VAE [71] aligning autoencoders with vision foundation models, and Cosmos [1] providing tokenizers across flexible compression regimes. In parallel, semantic-aligned encoders (DINOv2 [41], SigLIP [61], Qwen-VL [7, 6], V-JEPA 2.1 [38]) provide structured visual features, but their high dimensionality can make generative modeling unstable [54, 72]. Representation autoencoders (RAEs) address this by pairing frozen pretrained encoders with learned decoders [79, 60], enabling semantic latent spaces that support both visual understanding and generation [59]. However, high-dimensional RAE features can still suffer from off-manifold sampling and weak fine-geometry reconstruction [78], suggesting that RAEs do not simply replace VAEs but instead expose a tradeoff between pixel faithfulness and semantic abstraction [76]. For robotics, this tradeoff implies that the best latent space is not necessarily the one that reconstructs frames most faithfully, but the one that preserves action-relevant dynamics for prediction, planning, and policy evaluation.

### B.2 Action-Conditioned Diffusion Model

The world model is trained in the latent space of a frozen visual encoder. Let $o_{0:T-1}$ be a video clip, $a_{0:T-1}$ the corresponding action sequence, and $f_\phi$ the frozen encoder. We first form latents

$$z_{0:T-1} = f_\phi(o_{0:T-1}), \qquad z_t \in \mathbb{R}^{N\times D}, \tag{1}$$

where $N = h \times w$ is the number of spatial tokens and $D$ is the native encoder channel dimension. In the code tensors are stored as $h \times w \times D$, but the notation below flattens space to $N$ tokens. For adapter-based semantic encoders, $z_t$ is further compressed by the adapter $\alpha_\psi$ before being passed to the diffusion model,

$$\tilde{z}_t = \alpha_\psi(z_t), \qquad \tilde{z}_t \in \mathbb{R}^{N\times d}, \qquad d = 96. \tag{2}$$

The adapter and encoder are frozen during world model training; only the DiT parameters are optimized.

Table 5: **DiT size presets.** The hidden size, depth and the number of heads for each DiT size.

| Preset | Hidden Size $d$ | Depth | Heads | Head Dim $d/h$ |
|---|---|---|---|---|
| • DiT-S | 384 | 12 | 6 | 64 |
| • DiT-B | 768 | 12 | 12 | 64 |
| • DiT-L | 1024 | 24 | 16 | 64 |

All DiT runs in Table 4 use a DiT-L backbone with 24 layers, hidden size 1024, 16 attention heads, and $T$=10 frames. The context length is $H$=2, so the model conditions on $\tilde{z}_{0:H-1}$ and predicts the future block $\tilde{z}_{H:T-1}$ under actions $a_{0:T-1}$ and optional language $\ell$. The VAE latent has shape 32×32×16 and is patchified with DiT patch size $p$=2, while all semantic, Cosmos, and VA-VAE latents use a 16×16 token grid with $p$=1. Thus every row gives the DiT the same number of tokens per frame:

$$N = (h/p)(w/p) = 16 \cdot 16 = 256. \tag{3}$$

**This is the main reason high-dimensional semantic latents do not substantially increase DiT compute:** the transformer blocks operate on the same token count and hidden width, and the latent channel dimension only appears in the input patch projection and output prediction layer.

**Shallow-wide DDT head.** For high-dimensional representation latents, we also use a lightweight *shallow-wide* DDT head [67]. The DiT backbone remains unchanged. The shallow-wide head uses a 2048-dimensional readout width and keeps a minimal spatial refinement stage before the final patch prediction layer. This adds local spatial processing capacity at the output while leaving the main DiT backbone unchanged. As a result, the shallow head can improve the mapping from backbone features to high-dimensional representation with minimal increase in parameters.

**World model training hyperparameters.** All world models are trained on Bridge V2 clips resized to 256×256, with $T$=10 frames, $H$=2 history frames, frame skip 2, and 7-dimensional actions. Unless otherwise stated, the reported single-view runs use distributed data-parallel training on 4 H100 GPUs, per-GPU batch size 16 for DiT-S and 5 for DiT-L, bfloat16 autocast, and `torch.compile` [2]. The optimizer is AdamW [36] with learning rate (LR) of $10^{-4}$, betas $(0.9, 0.99)$, weight decay $2\times10^{-3}$, $\epsilon$=$10^{-8}$, and gradient clipping at global norm 1.0. We maintain an EMA copy of the DiT weights with decay 0.9995. The LR schedule is a linear warmup followed by cosine decay to 0.7 of the base LR. All runs use 3 LR warmup epochs and 100 total epochs.

**Flow matching.** The model is trained with the optimal-transport flow-matching objective [35]. For future frames $i \in \{H, \dots, T-1\}$, we sample $\tau_i \sim p(\tau)$, draw $\epsilon \sim \mathcal{N}(0, I)$, and linearly interpolate between data and noise:

$$\tilde{z}_{\tau_i,i} = (1-\tau_i)\tilde{z}_i + \tau_i\epsilon_i. \tag{4}$$

The DiT predicts the velocity field $v_\theta(\tilde{z}_\tau, \tau, a_{0:T-1}, \ell)$, and the target velocity is

$$u_i = \epsilon_i - \tilde{z}_i. \tag{5}$$

With clean history context, the training loss is

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{\tilde{z},\epsilon,\tau}\left[\sum_{i=H}^{T-1} \|v_\theta(\tilde{z}_{\tau,i}, \tau_i, a_{0:T-1}, \ell) - (\epsilon_i - \tilde{z}_i)\|_2^2\right]. \tag{6}$$

We only apply this loss to future frames. History frames are used as conditioning context with no diffusion noise ($\tau = 0$). During training, they however receive small Gaussian augmentation,

$$\tilde{z}^{\mathrm{ctx}}_{\mathrm{aug}} = \frac{\tilde{z}^{\mathrm{ctx}} + \sigma_h \eta}{\sqrt{1+\sigma_h^2}}, \qquad \eta \sim \mathcal{N}(0, I), \tag{7}$$

which prevents the model from overfitting to perfectly clean context latents.

**Dimension-dependent noise schedule shift.** For non-VAE latents, the timestep distribution is shifted as a function of the latent dimensionality seen by the DiT. Following Esser et al. [16] and Zheng et al. [79], we use the shift:

$$\gamma = \sqrt{\frac{(256/p^2)d}{4096}}, \qquad \tau' = \frac{\gamma\tau}{1+(\gamma-1)\tau}. \tag{8}$$

Here $d$ is the DiT input channel count after any adapter. This makes the noise level depend on the latent representation size rather than only on image resolution.

**Inference and causal attention.** All our world models carry out autoregressive inference in latent space. Given encoded history $\tilde{z}_{0:H-1}$, the sampler appends a Gaussian latent for the next frame and integrates the learned velocity field backward from $\tau$=1 to $\tau$=0 with 10 Euler steps [35, 16]:

$$\tilde{z}_{\tau_{j+1},t} = \tilde{z}_{\tau_j,t} - (\tau_j - \tau_{j+1}) v_\theta(\tilde{z}_{\tau_j,0:t}, \tau_j, a_{0:t}, \ell)_t. \tag{9}$$

The generated frame is then appended to the context and the process repeats for the desired horizon. Our temporal attention blocks are causal where each spatial token attends only to its own past states, following the causal video-transformer design used by VDT [37].

### B.3 Adapter

High-dimensional semantic encoders produce per-patch features $z \in \mathbb{R}^{N\times D}$ that are prohibitively expensive for the diffusion model to operate on directly. We pair them with an S-VAE adapter [78] that compresses $z$ to a compact latent $\tilde{z} \in \mathbb{R}^{N\times d}$ ($d \ll D$). The adapter $a_\psi$ comprises a Transformer encoder $g^{\mathrm{enc}}_\psi$, a per-token diagonal-Gaussian bottleneck, and a Transformer decoder $g^{\mathrm{dec}}_\psi$:

$$h = g^{\mathrm{enc}}_\psi(z), \tag{10}$$

$$(\mu, \log\sigma^2) = W_{\mu,\sigma^2}\, h, \tag{11}$$

$$\tilde{z} = \mu + \sigma \odot \xi, \qquad \xi \sim \mathcal{N}(0, I), \tag{12}$$

$$\hat{z} = g^{\mathrm{dec}}_\psi(\tilde{z}). \tag{13}$$

Both $g^{\mathrm{enc}}_\psi$ and $g^{\mathrm{dec}}_\psi$ consist of 3 Transformer blocks at dimension $D$, each followed by LayerNorm. The encoder appends a linear head $D \to 2d$ and the decoder prepends a linear head $d \to D$. We default to using 12 attention heads and FFN width of 3072.

The adapter training loss is:

$$\begin{aligned}\mathcal{L}_{\mathrm{adapter}} = \underbrace{\mathcal{L}_{\mathrm{MSE}}(z,\hat{z}) + \lambda_{\cos}\mathcal{L}_{\cos}(z,\hat{z}) + \lambda_{\mathrm{spec}}\mathcal{L}_{\mathrm{FFT}}(z,\hat{z})}_{\text{semantic reconstruction}} + \\ \lambda_{\mathrm{KL}}\, D_{\mathrm{KL}}(q_\psi(\tilde{z} \mid z)\,\|\,\mathcal{N}(0,I)) + \lambda_{\mathrm{pix}}\,\mathcal{L}_{\mathrm{pix}}(o,\hat{o}),\end{aligned} \tag{14}$$

where $\hat{o} = \mathrm{Dec}(\tilde{z})$ is the pixel-decoder reconstruction. $\mathcal{L}_{\mathrm{MSE}}$ and $\mathcal{L}_{\cos} = 1 - \cos(z,\hat{z})$ jointly enforce feature-space fidelity: MSE penalises magnitude errors while the cosine term preserves directional (semantic) structure. $D_{\mathrm{KL}}$ regularizes the approximate posterior $q_\psi(\tilde{z} \mid z) = \mathcal{N}(\mu, \sigma^2 I)$ toward a standard Gaussian prior. $\mathcal{L}_{\mathrm{FFT}}$ is an $\ell_1$ loss on 1-D FFT magnitudes along the spatial-token axis, penalizing loss of high-frequency structure through the bottleneck. $\mathcal{L}_{\mathrm{pix}} = \mathcal{L}_{\mathrm{MSE}}(o,\hat{o}) + \lambda_{\mathrm{LPIPS}}\mathcal{L}_{\mathrm{LPIPS}} + \lambda_{\mathrm{SSIM}}(1 - \text{MS-SSIM})$ grounds the compact latent in pixel space. Following Zhang et al. [78], we use $\lambda_{\mathrm{spec}}$=0.01, $\lambda_{\mathrm{LPIPS}}$=$\lambda_{\mathrm{SSIM}}$=0.5. During DiT training, $\alpha_\psi$ is frozen and applied deterministically ($\tilde{z} = \mu$) as a fixed projection into the compact latent space.

Table 6: **Training wall-clock and compute resources.** Times are measured wall-clock durations for the reported Bridge V2 training runs on 4 H100 GPUs and exclude one-time dataset staging operation. Note that all latent spaces roughly take the same training time due to the fixed number of token count

| Training run | Model size | Epochs | GPUs | Per-GPU batch | Precision | Wall-clock |
|---|---|---|---|---|---|---|
| Adapter + pixel decoder | S-VAE + CNN decoder | 200 | 4×H100 | 16 | bf16 | ∼55 h |
| World model | DiT-S | 100 | 4×H100 | 16 | bf16 | 6–7 h |
| World model | DiT-L | 80 | 4×H100 | 5 | bf16 | ∼34 h |

**Adapter training hyperparameters.** The encoder is frozen throughout adapter training. It is trained for 200 total epochs on Bridge V2, per-GPU batch size 16 for single-view training, and bfloat16 autocast. The optimizer is AdamW with betas $(0.9, 0.99)$ and weight decay $10^{-4}$. The base adapter learning rate is $10^{-4}$ for the single-view run; the pixel decoder uses a $3\times$ learning-rate multiplier when trained jointly. Multi-view adapter fine-tuning uses learning rate $5\times10^{-5}$ and lower per-GPU batch sizes because each sample contains three camera views. The KL coefficient is linearly warmed up for the first 20% of optimizer steps to $\lambda_{\mathrm{KL}}=10^{-4}$, while $\lambda_{\cos}=1$ and $\lambda_{\mathrm{pix}}=1$. LPIPS, when enabled, is evaluated in float32 after a 50k-sample perceptual warmup. Gradients for both the adapter and pixel decoder are clipped to norm 1.0.

### B.4 Pixel Decoder

The semantic encoders use the adapter pixel decoder for reconstruction. The pixel decoder maps compact latents $\tilde{z} \in \mathbb{R}^{N\times 96}$ to RGB observations:

$$\hat{o} = \mathrm{Dec}(\tilde{z}) = D_{\omega}^{\mathrm{pix}}(\tilde{z}). \tag{15}$$

Architecturally, it is an LDM-style convolutional decoder with two residual blocks per level, and a 4-head self-attention block at $16\times16$ resolution. For the S-VAE setup, the pixel decoder is trained on detached adapter latents with the pixel loss $\mathcal{L}_{\mathrm{pix}}$. As such, the pixel loss does not backpropagate into the adapter. The pixel reconstruction loss used in adapter training is

$$\mathcal{L}_{\mathrm{pix}} = \|\hat{o} - o\|_2^2 + \lambda_{\mathrm{LPIPS}}\mathcal{L}_{\mathrm{LPIPS}}(\hat{o}, o) + \lambda_{\mathrm{SSIM}}\left(1 - \text{MS-SSIM}(\hat{o}, o)\right). \tag{16}$$

In the S-VAE stage, the pixel decoder is trained on detached adapter latents, so pixel loss does not backpropagate into the adapter. The reported experiments use this S-VAE path rather than the older PS-VAE mode. For native semantic DiTs without an adapter in the diffusion model, visualization still uses the same surrogate path: native latent $\rightarrow$ adapter encoder $\rightarrow$ pixel decoder.

### B.5 Encoder-specific overhead

Table 4 summarizes parameter counts and compute. We split the parameter counts by frozen encoder, adapter, DiT, and decoder. GFLOPs are reported per frame for a single $256\times256$ frame by counting multiply-add as two separate floating-point operations. The total compute column adds encoder, adapter projection when used, one DiT velocity evaluation, and the decoder used for visualization/reconstruction. The differences in total GFLOPs in Table 4 are therefore mostly due to the frozen encoder and decoder, and not the DiT backbone itself. The DiT sees the same $N=256$ tokens per frame across all models, so increasing the semantic latent channel dimension mainly changes the input/output projections. In contrast, the encoders use different network families: VAE and VA-VAE are convolutional autoencoders operating over high-resolution feature maps, V-JEPA 2.1 and Web-DINO are ViT-style patch encoders [14], and SigLIP 2 is a larger, higher-capacity ViT-style vision model. Decoder compute also differs substantially: VAE uses its native convolutional decoder, VA-VAE uses a lighter convolutional decoder, and the semantic encoders use the adapter pixel decoder from a compact $16\times16$ latent grid. Thus the native 1024–1152D semantic rows have nearly the same DiT GFLOPs as their adapter-based $d_{96}$ counterparts.

Table 6 reports the measured training time and GPU configuration for the adapter/pixel-decoder stage and the DiT scaling runs.

### B.6 Inverse Dynamics Model (IDM)

The Inverse Dynamics Model (IDM) [57] is a patch-token Transformer trained to predict an action chunk $\hat{a}_{t:t+k-1} \in \mathbb{R}^{k \times d_a}$ from a window of $k+1$ consecutive encoder latents $(z_t, z_{t+1}, \ldots, z_{t+k})$, where each $z_t = f_\phi(o_t) \in \mathbb{R}^{N \times D}$ is the spatial patch grid produced by the frozen encoder $f_\phi$ directly, *i.e.*, no adapter $a_\psi$ is applied, so the IDM always operates in the native encoder channel space of dimension $D$. Each frame's $N = h \times w$ patch tokens are projected by a shared linear layer into a model-width embedding, augmented with factored temporal and spatial positional embeddings, and then flattened into a joint sequence of $(k+1) \cdot N$ tokens. A set of $k$ learned per-step class token (CLS) readout queries is prepended to this sequence; all tokens attend jointly through $L$ pre-norm Transformer blocks with scaled dot-product self-attention [65], and the final-layer representations of the $k$ CLS positions are decoded by a two-layer MLP head to the predicted action chunk $\hat{a}_{t:t+k-1}$. Following Tian et al. [57], we train each encoder-specific IDM on real encoded trajectories from Bridge V2 with Smooth-L1 loss.

The IDM serves as a probe of action recoverability for each encoder space $f_\phi \in \Phi$. After training, it is evaluated at horizons $k \in \{1, 4\}$ using Pearson $r$ between the predicted action chunk $\hat{a}_{t:t+k-1}$ and the ground-truth $a^*_{t:t+k-1}$, averaged over the $d_a$ continuous action dimensions. Critically, the same frozen IDM head is then applied without retraining to world model-generated latent pairs $(\hat{z}_t, \hat{z}_{t+k})$ from DiT rollouts of the same episodes. The Pearson $r$ of the real-WM gap thus measures generation-induced erasure of the action-discriminative geometry in the latent space, a form of degradation invisible to pixel-level metrics such as SSIM or LPIPS.

### B.7 VLA success classifier probe

The success classifier probe $s_\phi$ is a spatio-temporal Transformer trained on full latent trajectories $z_{0:T}$ from the SOAR dataset [81] to classify episode success $y \in \{0, 1\}$ given the language instruction $\ell$. Each trajectory's spatial latent grid is first downsampled to a $4 \times 4$ super-patch grid via adaptive average pooling, yielding $P{=}16$ spatial tokens per frame, and linearly projected to a shared model width of 384. Factored temporal and spatial positional embeddings are added in place, producing a token tensor of shape $(T \times P)$; a learned CLS token is then prepended. Each of the six blocks of the success probe applies three sequential sub-operations with pre-norm and residual connections: a) spatial self-attention within each frame independently over the $P$ patch tokens, b) temporal self-attention across the $T$ frames independently per patch position, and c) cross-attention from all video tokens to the frozen SigLIP 2 token sequence encoding $\ell$, followed by a SwiGLU FFN. After the final RMSNorm, the mean of the $T \times P$ patch token representations is passed through a linear head to produce a binary logit $\hat{y}$.

The probe is trained with binary cross-entropy on SOAR episodes, with the encoder $f_\phi$, adapter $a_\psi$, and SigLIP 2 text encoder all frozen; only the parameters of $s_\phi$ are updated. Instruction-mismatch negatives (episodes paired with a language instruction drawn from a different task family) are mixed in to force the above cross-attention mechanism to genuinely ground success in the video content rather than ignoring $\ell$. Checkpoints are selected by balanced accuracy with ROC-AUC as the tie-breaker, accounting for SOAR's 1:2 success-to-failure class imbalance. At evaluation, the same frozen $s_\phi$ is applied without retraining to world model-generated latent trajectories $\hat{z}_{0:T}$ from DiT rollouts of the same SOAR episodes. The drop in balanced accuracy from **Enc. Acc** to **WM Acc** measures the semantic drift, *i.e.*, the degree to which the transition model $p_\theta$ degrades task-outcome separability in latent space over the full rollout horizon, a signal invisible to per-step action metrics.

## C Evaluation metrics

### C.1 Planning and downstream policy performance

We evaluate planning and policy performance through three complementary sub-protocols: CEM-based latent controllability, VLA-in-the-loop closed-loop success, and robustness under distribution shift. Throughout, $a_t \in \mathbb{R}^{d_a}$ is the action vector with seven degrees of freedom, $a^*_{t:t+k-1}$ is the ground-truth $k$-step action sequence, and $\tilde{z}_t$ is the compact latent on which the DiT $p_\theta$ operates.

**A) CEM action controllability.** We evaluate whether a trained world model preserves action information by asking whether actions can be recovered from its latent dynamics. Given a held-out transition window with two real context latents, $(\tilde{z}_t, \tilde{z}_{t+1})$, ground-truth action sequence $a^*_{t+1:t+k}$, and target future latents $\tilde{z}^*_{t+2:t+k+1}$, we solve

$$a^{\text{plan}}_{t+1:t+k} = \arg\min_{a_{t+1:t+k}} \frac{1}{k} \sum_{j=1}^{k} \left\| p^{(j)}_\theta(\tilde{z}_t, \tilde{z}_{t+1}, a_{t+1:t+k}) - \tilde{z}^*_{t+1+j} \right\|_2^2. \tag{17}$$

Here $p^{(j)}_\theta$ denotes the $j$th autoregressive latent prediction from the world model. We report results for $k \in \{1, 4\}$ using 100 held-out windows per model.

The optimization in Eq. (17) uses the cross-entropy method (CEM) [49]. For each transition window, CEM maintains a diagonal Gaussian over the optimized action coordinates for all $k$ steps. In the reported runs, we use a population of 400 candidate action sequences, 5 CEM iterations, and 50 elites per iteration, i.e. an elite fraction of $0.125$. The sampling distribution is initialized with mean $a^*_{t+1:t+k}$ on the searched coordinates and standard deviation equal to one quarter of the action range for each searched coordinate. After each iteration, the Gaussian mean and standard deviation are set to the empirical mean and standard deviation of the elite set.

Each CEM candidate is evaluated with one latent rollout sample. The diffusion sampler uses the same inference setting as evaluation, with 10 flow-matching Euler steps per predicted latent frame. To make the CEM objective deterministic for a given transition, we sample one Gaussian rollout-noise tensor per transition window and reuse it for all candidates and all CEM iterations. Thus the world-model rollout is stochastic across evaluation windows through the sampled diffusion noise, but the optimizer sees a fixed objective within each window. For $k > 1$, candidates are evaluated by a joint autoregressive rollout: after the first predicted latent, the prediction is appended to the context and used to predict the next latent under the next candidate action.

We compute the **CEM error** from the recovered action sequences: $\frac{1}{k}\sum_{j=1}^{k} \|a^{\text{plan}}_{t+j,S} - a^*_{t+j,S}\|_2$, averaged over transitions, where $S$ is the set of searched action dimensions. Lower error indicates that the world-model latent dynamics are more action-sensitive under the CEM inversion test.

**B) VLA-in-the-loop closed-loop success.** We roll out OpenVLA-7B [28] inside each world model for 50-step episodes across 20 Bridge V2 test episodes with 8 independent trials per episode (*i.e.*, $N = 80$ total rollouts ). Each rollout video is scored by two VLMs, InternVL-3.5-14B [68] and Qwen-3.6-27B [46] using 16 tail-biased frames sampled from the rollout. We use these to compute the following closed-loop success metrics:

- Consensus success rate (**Consensus SR**) reports the fraction of trials scored as a success by *both* raters simultaneously: $\text{CSR} = \frac{1}{N}\sum_i \mathbf{1}[\text{score}_i^{\text{InternVL}} \geq 0.5 \wedge \text{score}_i^{\text{QwenVL}} \geq 0.5]$. Requiring agreement from both raters reduces false positives from any single rater's miscalibration.
- **Borda rank** is the sum of rank positions across both raters within each DiT-size group: $\text{Borda} = r_{\text{InternVL}} + r_{\text{QwenVL}}$, where $r_{\text{InternVL}}$ and $r_{\text{QwenVL}}$ are the ordinal ranks of the model by SR-InternVL and SR-QwenVL respectively: rank 1 being the best. This is an ordinal measure robust to rater calibration drift and a lower score is better.

**C) VLM interaction-quality rubric.** Each rollout is additionally scored by InternVL 3.5 [68] on a structured rubric with three independent sub-scores on a 1–5 integer scale, then averaged across the $N$ trials [51]. It is rated by a VLM using the prompt described in Sec. C.5.

- Interaction quality score (**IQ score**↑) measures the plausibility of robot–object contact, including whether grasps, pushes, and force transfers look realistic and avoid interpenetration artifacts. This helps capture whether the world model renders credible manipulation dynamics without requiring pixel-level ground truth.
- Instruction following (**Instr. follow**↑) is the degree to which the rollout visually executes the language instruction $\ell$ (e.g., grasping the correct object, moving in the specified direction). Instruct follow is comple-

mentary to binary SR in the sense that it captures partial progress on episodes where neither judge counts the rollout as a full success.

**D) Out-of-Distribution (OOD) robustness.** We re-run a subset of 10 tasks from the 20 used for calculating VLA SR, 8-trial setup under two independent perturbations. Distractor-object (OOD distractor) rollouts add OOD objects to the scene as described in Sec. C.5, while OOD-instruction rollouts replace the language instruction $\ell$ with a semantically unrelated instruction drawn from a different Bridge V2 task family. Success rates under perturbation use the mean of the two per-rater SRs:

- OOD SR Distractor: the per-rater mean SR under distractor objects.
- OOD SR instruction: the per-rater mean SR under the substituted instruction.

### C.2 Pixel fidelity and scene geometry

Action faithfulness is a necessary but not sufficient condition for world modeling, *e.g.*, a model that steers correctly yet generates physically implausible scenes will still mislead a policy that relies on visual observations. We thus evaluate decoded rollout quality across three categories visual quality, content consistency, and motion quality each containing *reference-based* metrics that compare generated frames $\hat{o}_t$ to paired ground-truth frames $o_t^*$, and *reference-free* perceptual metrics [51] that score generated clips without a ground-truth counterpart. All metrics are computed over 1,000 test episodes.

**A) Visual quality.** Reference-based metrics include:

- **PSNR**↑ measures the peak signal-to-noise ratio $10\log_{10}(1/\mathrm{MSE}(\hat{o}_t, o_t^*))$, averaged over frames and episodes. This helps quantify pixel-level reconstruction accuracy but not the perceptual structure.
- **SSIM**↑ measures structural similarity [69] between $\hat{o}_t$ and $o_t^*$, computed on luminance with a local window. Captures structural and contrast coherence that PSNR misses.
- **LPIPS**↓ measures the learned Perceptual Image Patch Similarity [77] using AlexNet [30] features. LPIPS correlates better with human perceptual judgments than pixel-level metrics, penalizing blurry or structurally incorrect generations even when MSE is low.
- **FID**↓ quantifies the Fréchet Inception Distance [24] between the distribution of generated and ground-truth frames, computed from InceptionV3 2048-D features [55]. FID measures the population-level gap between generated and real frame distributions, capturing systematic biases that per-frame metrics average away.

Reference-free metrics are borrowed from Shang et al. [51] and include:

- **Image quality**↑ measures the MUSIQ [27] multi-scale image quality score, normalized to $[0, 1]$. This helps quantify the perceptual quality of individual frames using a model trained on human quality ratings, without requiring a ground-truth reference.
- **Aesthetic quality**↑ uses the LAION aesthetic predictor score [50], normalized to $[0, 1]$ from a raw $[0, 10]$ scale. This helps capture the compositional and stylistic appeals of generated frames independently of content accuracy.
- **JEPA similarity** ↑ measures the maximum mean discrepancy (MMD) between feature distributions extracted from JEPA [3] to provide evaluation results that better align with human perception.

**B) Motion quality.** Reference-based metrics include:

- **FVD**↓ measures the Fréchet Video Distance [63] computed from ResNet-3D features on 16-frame clips. FVD helps extend FID to the temporal domain, thus capturing spatiotemporal distribution quality of full video clips rather than individual frames.
- **t-LPIPS**↓ uses RAFT [56] to estimate the optical flow $\mathbf{u}_{t-1\to t}$ on the ground-truth frames. Both generated and ground truth (GT) frames are then warped with this shared flow. t-LPIPS is the mean absolute difference between the per-step LPIPS of the flow-warped generated video and the flow-warped GT video. Using GT

flow as a shared reference decouples temporal dynamics quality from content. Here, a low score signifies the model's frame-to-frame motion pattern matches ground truth.

- **PCK coverage**↑ uses CoTracker [26] to track a $16\times16$ grid of query points placed on the first context frame through the generated video. PCK coverage is the mean fraction of these query points that remain visible (tracked with high confidence) at each rollout step. A drop across steps indicates that the generated video causes points to leave the frame or become untrackable, which implies geometric instability.

Reference-free metrics are borrowed from Shang et al. [51] and include:

- **Dynamic degree**↑ measures the fraction of inter-frame pairs in a generated clip where RAFT-estimated optical flow magnitude exceeds a threshold $\tau$=6 pixels. A near-zero value indicates a nearly static rollout, which is unlikely to be action-faithful regardless of pixel quality.
- **Flow score**↑ quantifies the mean magnitude of the top-5% of optical flow vectors across all inter-frame pairs in a generated clip. This helps capture the strength of dominant motion events, complementing dynamic degree which only measures their frequency.

**C) Reconstruction ceiling.** For each encoder, all reference-based metrics are additionally computed on *reconstructed* frames, *i.e.*, real observations encoded and decoded without any DiT. This gives us a per-encoder upper bound. The gap $\Delta$ is the difference between the world model score and this ceiling, isolating the quality loss attributable to the transition model rather than the decoder. A large gap indicates that the DiT struggles to generate in-distribution latents while a small gap implies that the encoderdecoder path is not the bottleneck.

### C.3 Latent representation quality

**A) Action Recoverability.** A world model can score well on PSNR/SSIM yet use an encoder that never encoded action information to begin with, or use a good encoder but a DiT that overlooks the action-discriminative geometry during denoising. Action recoverability metrics seek to address these and include the following reference-based measures:

- **IDM Pearson** $r$ **(Encoder)** uses an Inverse Dynamics Model (IDM) head [57] trained on consecutive frozen encoder latent pairs $(z_t,\, z_{t+k})$ from Bridge V2 to predict an action chunk $\hat{a}_{t:t+k-1} \in \mathbb{R}^{k\times7}$ for horizon $k \in \{1, 4\}$. Pearson $r$ is then computed by averaging over the six continuous action dimensions on held-out real encoded frames, establishing the maximum step-level action information linearly accessible from each encoder space.
- **IDM Pearson** $r$ **(WM)** applies the same frozen IDM (trained on real latents) to world model generated latent pairs $(\hat{z}_t,\, \hat{z}_{t+k})$ from DiT rollouts of the same episodes. A small Real–WM $r$ difference confirms the transition model faithfully preserves action-relevant latent geometry during generation while a large difference exposes degradation invisible to pixel metrics. A large gap between Real and WM $r$ indicates generation-induced erasure of action-distinguishing structure even when decoded pixels look faithful.

**B) Success classifier Accuracy or Success Separability.** We seek to measure whether the world model's generated latent trajectories retain enough task-outcome structure for a frozen success classifier to distinguish successful from failed episodes, *i.e.*, the DiT preserves semantic meaning over a full rollout and not just local action geometry. Semantic fidelity includes the following reference-based metrics that require ground-truth success/failure labels:

- **Enc. Acc** signify the encoder ceiling, where a factored spatial–temporal attention probe $g_\phi$, conditioned on frozen SigLIP 2 text tokens, is trained on real encoder latent trajectories $z_{0:T}$ from SOAR [81] to classify task success given the language instruction. Balanced accuracy on held-out real-encoded trajectories establish the probe ceiling, *i.e.*, the maximum task-success information preserved in each encoder space.
- **WM Acc.** applies the frozen probe $g_\phi$ is applied without retraining to full world model generated latent rollouts of the same episodes. Lower WM Acc relative to Enc. Acc reveals *semantic drift*: the generated trajectory has lost task-outcome separability even when per-step action signals remain partially intact.

### C.4 VLA-based evaluations

Table 7: **OOD-instruction evaluation pairs.** Each original instruction is paired with a single semantically-related but behaviorally distinct instruction. Variations span four types: action reversal (same scene, opposite action), action + target change, spatial relation change, and target location change.

| Original instruction | OOD instruction | Variation |
|---|---|---|
| close oven | open the oven | *Action reversal* |
| open the drawer | close the drawer | *Action reversal* |
| fold the cloth from the bottom to the top | unfold the cloth flat | *Action reversal* |
| sweep into pile | scatter the pile across the table | *Action reversal* |
| pick up sponge and wipe plate | drop the sponge into the sink | *Action + target change* |
| Move the can behind the blue fork | place the can on top of the blue fork | *Spatial relation* |
| pick up blue towel from the grey thing and placed it to the right of the white basket | put the blue towel inside the white basket | *Spatial relation* |
| put the covering lid on top of the silver pot | put the lid inside the silver pot | *Spatial relation* |
| moved the blue scrubber onto the lower right burner | move the blue scrubber to the upper left burner | *Spatial location* |
| place the silver pot in the middle of the table | place the silver pot in the sink | *Target location* |

We manually pick the set of 20 tasks present in Table 11 to have a good mix of task difficulties as well as task diversities from the Bridge V2 test set. The tasks involve instructions like pick and place, opening/closing, interacting with non-rigid objects like clothes, and tasks that require precise arm and gripper control. We use Claude Opus 4.7 to generate the OOD instructions given the original task instruction in Table 7. These OOD instructions also span several variations.

### C.5 VLM prompts

We list the exact prompts we used to create the out of distribution distractor images, score the VLA policy trajectories, and to score the interaction quality and related metrics. We provide a summary of the full prompt from Shang et al. [51] for the latter here. We chose a subset of 10 tasks equally sampled from the difficulty levels in Table. 11 and use ChatGPT Images 2.0 model with the distractor prompt given below to generate the initial frame with OOD objects added to the scene.

**Distractor Image Editing**

**Use:** Text-guided distractor insertion for OOD distractor objects

Exact prompt template:

```
This is an initial observation for a robotics task {task_instruction}.
Modify this image by adding distraction objects in the scene in a natural
way without moving or changing any objects in the original scene.

Requirements:
- The robotic arm should be visible.
- With the distractors, the task {task_instruction} should remain
achievable.
```

**Episode Success/Failure Scoring**

**Use:** Online rollout scoring used by policy-in-the-loop evaluation

Prompt structure:

```
Here is a sequence of frames from a robot policy which has been
rolled out in a video-generation-based world model.  I need your help
determining whether the policy is successful.  How successfully does the
robot complete the following task?

Instruction:  {instruction}

Score rubric:
0 = Failure
0.5 = Partial (optional, when partial criteria are provided)
```

```
1 = Success
Provide brief reasoning (2-3 sentences). Then output exactly one final
line: Final Score: X
```

The binary version uses only `0/1`. The partial-credit version adds `0.5` when a `partial_criteria` string is present.

**Interaction Quality / Perspectivity / Instruction Following**

**Use:** Multi-dimensional VLM judge rubric for paper-style interaction-quality metrics.

Prompt summary:

- Evaluates **three** dimensions on a 1–5 Likert scale: Interaction Quality, Perspectivity, and Instruction Following.
- Scene prior is explicit: tabletop or counter-top *robotic arm* manipulation, not human-hand videos.
- Includes a hard hallucination check: if the video shows human hands instead of robotic arms, Instruction Following should be scored at most 2.
- Requires the model to base judgments only on visible evidence in the sampled frames and to consider temporal coherence.
- Output is forced to a single JSON object with exactly three top-level keys:

```
{"Interaction_Quality": {"score": 1-5, "reason": "..."},
"Perspectivity": {"score": 1-5, "reason": "..."},
"Instruction_Following": {"score": 1-5, "reason": "..."}}
```

We rated all trajectories using three open-source, but strong Vision Language Models (VLMs), InternVL3.5-14B [68], Qwen3.6-27B [46], and Qwen3.5-9B [45] with the same scoring prompt and sampled frames. We sampled 16 frames from each episodes, with 10 sampled uniformly throughout the video and 6 sampled uniformly from the second half of the episode. We did this since the ending of a trajectory often has more task success relevant information. We then calculate Cohen's kappa $\kappa$ to measures the agreement between each of the VLM raters (Fig. 8), and find that InternVL3.5-14B and Qwen3.6-27B are in moderate agreement. Thus we chose the consensus rating from these two VLMs for our success rate figures. We also verify that the main trend is supported by non-VLM metrics: CEM, IDM, success probes, and visual/geometric metrics.

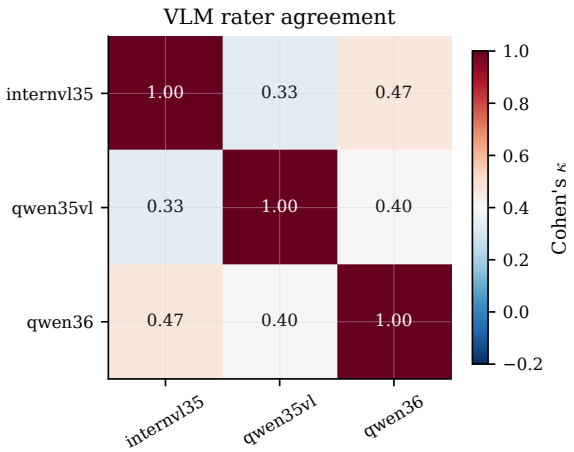



Figure 8: **The Cohen's kappa** for inter-VLM rater agreement. Given the higher agreement between InternVL 3.5 and Qwen 3.6, we choose these as our VLM judges for policy-in-the-loop task success experiments.

## D Additional Results

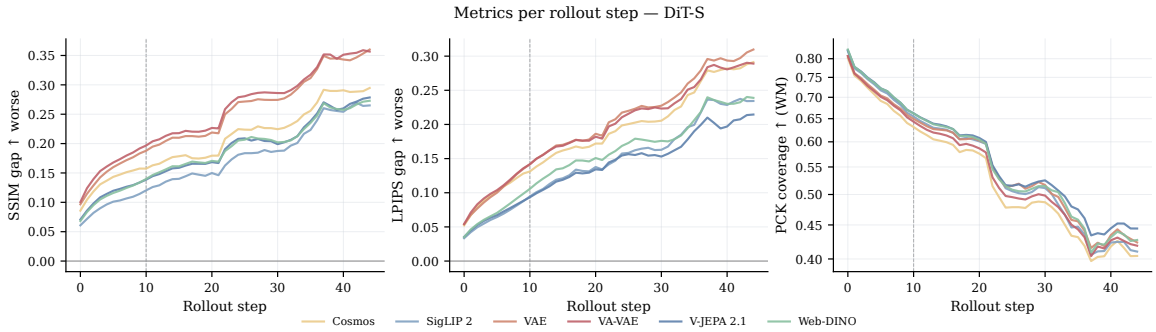



Figure 9: **SSIM gap, LPIPS gap, and PCK coverage** over 45 rollout steps. While all encoders show a strictly increasing SSIM/LPIPS gap over the full rollout due to compounding errors (each autoregressive step feeds back slightly corrupted predictions as context), semantic latent spaces from SigLIP2, V-JEPA 2.1 and Web-DINO remain particularly competitive when forced to extrapolate beyond the 10-frame horizon length seen during training. Conversely, PCK coverage remains the highest for semantic encoders.

## D.1 Visual performance across DiT backbone sizes

Table 8: **Reconstruction quality across DiT sizes: S, B, and L.** Each cell for PSNR, SSIM, LPIPS, t-LPIPS, FID, and FVD shows the WM value with the gap to its encoder's reconstruction ceiling in parentheses (lower is closer to the ceiling). **Best** and runner-up within each size group; the WM value and gap are highlighted independently.

| Encoder | Reconstruction fidelity | | | | | Generative quality | |
|---|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PCK coverage ↑ | t-LPIPS ↓ | FID ↓ | FVD ↓ |
| **DiT-S** | | | | | | | |
| • VAE | 17.43 (16.65) | 0.688 (0.251) | 0.218 (0.207) | 0.565 | 0.0264 (0.0252) | 17.43 (16.63) | 6.8 (6.5) |
| • VA-VAE | 16.98 (13.24) | 0.633 (0.256) | 0.226 (0.201) | 0.559 | 0.0253 (0.0228) | 15.49 (13.24) | 8.5 (7.0) |
| • Cosmos | 16.97 (10.06) | 0.608 (0.210) | 0.245 (0.197) | 0.544 | 0.0223 (0.0184) | 16.95 (11.37) | 8.2 (4.5) |
| • V-JEPA 2.1 | 18.10 (11.09) | 0.725 (0.190) | **0.176** (0.141) | **0.580** | 0.0197 (0.0161) | 6.77 (2.75) | 5.5 (2.3) |
| • V-JEPA 2.1$_{96}$ | 18.20 (10.99) | 0.729 (0.186) | 0.179 (0.143) | 0.575 | 0.0212 (0.0176) | 6.30 (2.27) | **5.2** (2.1) |
| • Web-DINO | 17.42 (10.87) | 0.722 (0.188) | 0.199 (0.160) | 0.575 | 0.0234 (0.0190) | 7.63 (3.37) | 6.7 (3.1) |
| • Web-DINO$_{16}$ | 17.82 (**8.54**) | 0.711 (0.171) | 0.196 (0.139) | 0.563 | 0.0200 (0.0144) | 8.37 (2.55) | 7.6 (2.4) |
| • Web-DINO$_{64}$ | **18.26** (9.54) | 0.738 (0.168) | 0.196 (**0.124**) | 0.575 | **0.0185** (**0.0118**) | 14.18 (**1.39**) | 10.9 (**1.5**) |
| • Web-DINO$_{96}$ | 17.99 (10.26) | 0.728 (0.180) | 0.181 (0.142) | 0.572 | 0.0195 (0.0151) | **6.00** (2.16) | 5.5 (2.2) |
| • Web-DINO$_{256}$ | 17.63 (10.74) | 0.725 (0.187) | 0.214 (0.148) | 0.574 | 0.0231 (0.0170) | 14.25 (2.16) | 10.4 (2.1) |
| • SigLIP 2 | 17.48 (9.29) | 0.713 (0.181) | 0.205 (0.156) | 0.555 | 0.0228 (0.0173) | 7.86 (2.89) | 6.9 (2.4) |
| • SigLIP 2$_{96}$ | 18.06 (8.69) | **0.738** (**0.152**) | 0.179 (0.131) | 0.578 | 0.0223 (0.0168) | 6.88 (2.40) | 6.0 (1.8) |
| **DiT-B** | | | | | | | |
| • VAE | 17.47 (16.60) | 0.682 (0.257) | 0.206 (0.196) | 0.565 | 0.0236 (0.0226) | 10.64 (9.84) | 5.1 (4.7) |
| • V-JEPA 2.1 | **18.43** (10.81) | **0.740** (**0.178**) | **0.171** (**0.135**) | **0.584** | **0.0205** (**0.0169**) | 7.31 (2.81) | 5.9 (2.5) |
| • V-JEPA 2.1$_{96}$ | 18.06 (11.07) | 0.726 (0.186) | 0.185 (0.150) | 0.573 | 0.0206 (0.0171) | 5.99 (**2.40**) | **5.0** (**2.1**) |
| • Web-DINO | 17.76 (**10.40**) | 0.716 (0.185) | 0.190 (0.151) | 0.571 | 0.0233 (0.0189) | **5.96** (2.81) | 5.5 (2.7) |
| **DiT-L** | | | | | | | |
| • VAE | 18.44 (15.63) | 0.729 (0.210) | 0.168 (0.157) | 0.575 | 0.0202 (0.0191) | **5.35** (4.56) | **3.5** (3.1) |
| • Cosmos | 18.01 (9.02) | 0.657 (0.162) | 0.186 (0.138) | 0.563 | 0.0199 (0.0159) | 9.23 (3.65) | 6.5 (2.9) |
| • V-JEPA 2.1 | 18.53 (10.70) | 0.741 (0.177) | 0.172 (0.136) | 0.583 | 0.0195 (0.0159) | 6.94 (2.44) | 5.4 (1.9) |
| • V-JEPA 2.1$_{96}$ | **18.65** (10.54) | **0.743** (0.171) | **0.165** (0.130) | **0.584** | 0.0201 (0.0166) | 6.19 (2.16) | 5.2 (2.1) |
| • Web-DINO | 17.72 (10.56) | 0.729 (0.181) | 0.192 (0.153) | 0.581 | 0.0219 (0.0176) | 6.92 (2.65) | 6.0 (2.5) |
| • Web-DINO$_{96}$ | 18.62 (9.62) | 0.741 (0.154) | 0.189 (**0.117**) | 0.577 | **0.0189** (**0.0123**) | 14.26 (**1.33**) | 13.1 (**1.3**) |
| • SigLIP 2 | 17.94 (8.83) | 0.730 (0.163) | 0.188 (0.140) | 0.581 | 0.0207 (0.0153) | 7.57 (2.60) | 6.7 (2.2) |
| • SigLIP 2$_{96}$ | 18.30 (**8.45**) | 0.743 (**0.147**) | 0.171 (0.123) | 0.580 | 0.0193 (0.0138) | 6.74 (2.24) | 5.8 (1.6) |

Table 9: **World Arena perceptual metrics across DiT sizes: S, B, and L.** **Best** and runner-up within each size group.

| Encoder | Quality | | Frame consistency | | | Motion | | Reference-based | |
|---|---|---|---|---|---|---|---|---|---|
| | Image quality ↑ | Aesthetic quality ↑ | Subject consist. ↑ | Background consist. ↑ | Photometric consist. ↓ | Dyn. degree ↑ | Flow score ↑ | Depth AbsRel ↓ | JEPA sim. ↑ |
| **DiT-S** | | | | | | | | | |
| VAE | **0.592** | 0.467 | 0.810 | 0.950 | 96.26 | 0.767 | 1.186 | 0.390 | 0.871 |
| VA-VAE | 0.585 | 0.464 | 0.817 | 0.949 | 94.93 | 0.765 | 1.204 | 0.455 | 0.783 |
| Cosmos | 0.558 | 0.463 | 0.793 | 0.946 | **73.29** | 0.813 | 1.511 | 0.638 | 0.517 |
| V-JEPA 2.1 | 0.578 | 0.473 | 0.841 | 0.955 | 80.49 | 0.832 | 1.587 | 0.404 | 0.929 |
| V-JEPA $2.1_{96}$ | 0.579 | **0.474** | 0.841 | 0.955 | 76.30 | **0.843** | **1.653** | 0.363 | 0.928 |
| Web-DINO | 0.576 | 0.472 | 0.849 | 0.957 | 94.03 | 0.794 | 1.408 | **0.350** | 0.938 |
| Web-$\text{DINO}_{16}$ | 0.546 | 0.469 | 0.838 | 0.952 | 76.85 | 0.824 | 1.532 | 0.399 | 0.905 |
| Web-$\text{DINO}_{64}$ | 0.575 | 0.466 | 0.854 | 0.960 | 84.95 | 0.823 | 1.532 | 0.358 | 0.774 |
| Web-$\text{DINO}_{96}$ | 0.574 | 0.473 | 0.841 | 0.955 | 76.82 | 0.835 | 1.634 | 0.375 | **0.944** |
| Web-$\text{DINO}_{256}$ | 0.581 | 0.467 | **0.861** | **0.961** | 103.86 | 0.782 | 1.325 | 0.357 | 0.785 |
| SigLIP 2 | 0.566 | 0.471 | 0.839 | 0.953 | 74.93 | 0.827 | 1.602 | 0.394 | 0.931 |
| SigLIP $2_{96}$ | 0.573 | 0.472 | 0.843 | 0.955 | 77.30 | 0.827 | 1.547 | 0.372 | 0.938 |
| **DiT-B** | | | | | | | | | |
| VAE | **0.591** | 0.471 | 0.813 | 0.951 | **79.77** | **0.824** | 1.520 | 0.434 | 0.915 |
| V-JEPA 2.1 | 0.582 | **0.474** | 0.847 | **0.958** | 87.57 | 0.812 | 1.454 | **0.324** | 0.923 |
| V-JEPA $2.1_{96}$ | 0.577 | 0.474 | 0.845 | 0.957 | 82.82 | 0.823 | **1.521** | 0.381 | 0.928 |
| Web-DINO | 0.577 | 0.473 | **0.847** | 0.957 | 86.57 | 0.815 | 1.493 | 0.342 | **0.939** |
| **DiT-L** | | | | | | | | | |
| VAE | **0.598** | **0.475** | 0.827 | 0.952 | 75.16 | **0.844** | 1.635 | **0.281** | **0.980** |
| Cosmos | 0.578 | 0.469 | 0.817 | 0.952 | **71.22** | 0.843 | **1.650** | 0.465 | 0.760 |
| V-JEPA 2.1 | 0.578 | 0.474 | 0.844 | 0.956 | 80.05 | 0.832 | 1.573 | 0.330 | 0.926 |
| V-JEPA $2.1_{96}$ | 0.581 | 0.474 | 0.842 | 0.956 | 81.48 | 0.831 | 1.558 | 0.346 | 0.929 |
| Web-DINO | 0.573 | 0.472 | 0.847 | 0.957 | 84.57 | 0.823 | 1.557 | 0.343 | 0.945 |
| Web-$\text{DINO}_{96}$ | 0.578 | 0.466 | **0.852** | **0.959** | 79.79 | 0.833 | 1.568 | 0.352 | 0.709 |
| SigLIP 2 | 0.569 | 0.472 | 0.845 | 0.956 | 79.16 | 0.822 | 1.562 | 0.344 | 0.937 |
| SigLIP $2_{96}$ | 0.573 | 0.472 | 0.844 | 0.956 | 76.98 | 0.830 | 1.580 | 0.326 | 0.937 |

### D.2 Policy performance across DiT backbone sizes

Table 10: **Policy and behavioral metrics for different DiT sizes:** small (S), base (B), and large (L). **Best** and runner-up within each size group. In-distribution (ID) SR: InternVL3.5 on the 10 episodes shared with OOD evaluations. OOD SR: InternVL3.5 only. Borda rank (lower = better) aggregates InternVL3.5-14B, and Qwen3.6-27B rankings. Muted ± terms show one standard deviation averaged over episode for SR and CEM metrics.

| Encoder | VLA SR | | Interaction quality | | PCK | OOD robustness | | | CEM error | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Cons. SR ↑ | Borda rank ↓ | IQ score ↑ | Instruction follow ↑ | PCK coverage ↑ | ID SR ↑ | OOD SR distractor ↑ | OOD SR instruction ↑ | k=1 ↓ | k=4 ↓ |
| **DiT-S** | | | | | | | | | | |
| • VAE | 0.169 ±0.030 | 31 | 3.26 | 3.48 | 0.719 | 0.375 ±0.054 | 0.287 ±0.051 | 0.200 ±0.045 | 0.111 ±0.009 | 0.612 ±0.023 |
| • VA-VAE | 0.175 ±0.030 | 28 | 3.22 | 3.42 | 0.715 | 0.350 ±0.053 | 0.250 ±0.048 | 0.200 ±0.045 | 0.097 ±0.005 | 0.543 ±0.023 |
| • Cosmos | 0.244 ±0.034 | 20 | 3.32 | 3.51 | 0.707 | 0.425 ±0.055 | 0.362 ±0.054 | 0.275 ±0.050 | 0.112 ±0.009 | 0.661 ±0.033 |
| • V-JEPA 2.1 | 0.344 ±0.038 | **7** | 3.43 | 3.78 | 0.735 | 0.600 ±0.055 | 0.575 ±0.055 | **0.400** ±0.055 | 0.084 ±0.008 | **0.424** ±0.014 |
| • V-JEPA 2.1$_{96}$ | **0.362** ±0.038 | 9 | **3.52** | 3.84 | 0.735 | 0.600 ±0.055 | 0.537 ±0.056 | 0.250 ±0.048 | 0.089 ±0.007 | 0.548 ±0.017 |
| • Web-DINO | 0.212 ±0.032 | 26 | 3.34 | 3.58 | 0.735 | 0.550 ±0.056 | 0.512 ±0.056 | 0.250 ±0.048 | 0.090 ±0.007 | 0.474 ±0.026 |
| • Web-DINO$_{16}$ | 0.256 ±0.035 | 13 | 3.51 | **3.85** | 0.721 | 0.500 ±0.056 | 0.500 ±0.056 | 0.300 ±0.051 | 0.104 ±0.008 | 0.555 ±0.020 |
| • Web-DINO$_{64}$ | 0.281 ±0.036 | 16 | 3.34 | 3.50 | 0.734 | 0.550 ±0.056 | 0.487 ±0.056 | 0.325 ±0.052 | — | — |
| • Web-DINO$_{96}$ | 0.300 ±0.036 | 13 | 3.44 | 3.77 | 0.732 | 0.600 ±0.055 | 0.512 ±0.056 | 0.275 ±0.050 | 0.090 ±0.007 | 0.531 ±0.025 |
| • Web-DINO$_{256}$ | 0.194 ±0.031 | 21 | 3.31 | 3.56 | **0.735** | 0.512 ±0.056 | 0.500 ±0.056 | 0.287 ±0.051 | — | — |
| • SigLIP 2 | 0.325 ±0.037 | 10 | 3.43 | 3.58 | 0.730 | 0.537 ±0.056 | 0.500 ±0.056 | 0.263 ±0.049 | **0.082** ±0.006 | 0.523 ±0.030 |
| • SigLIP 2$_{96}$ | 0.331 ±0.037 | 16 | 3.42 | 3.71 | 0.731 | **0.625** ±0.054 | **0.588** ±0.055 | 0.312 ±0.052 | 0.086 ±0.005 | 0.537 ±0.026 |
| **DiT-B** | | | | | | | | | | |
| • VAE | 0.256 ±0.035 | 11 | 3.31 | 3.62 | 0.723 | 0.463 ±0.056 | 0.438 ±0.055 | 0.225 ±0.047 | 0.113 ±0.010 | — |
| • V-JEPA 2.1 | 0.319 ±0.037 | **4** | 3.51 | **3.77** | **0.739** | **0.625** ±0.054 | 0.475 ±0.056 | **0.325** ±0.052 | 0.096 ±0.008 | — |
| • V-JEPA 2.1$_{96}$ | **0.325** ±0.037 | 6 | **3.52** | 3.69 | 0.737 | 0.575 ±0.055 | 0.525 ±0.056 | 0.200 ±0.045 | 0.097 ±0.009 | — |
| • Web-DINO | 0.287 ±0.036 | 8 | 3.44 | 3.75 | 0.736 | 0.600 ±0.055 | **0.550** ±0.056 | 0.300 ±0.051 | **0.084** ±0.006 | — |
| **DiT-L** | | | | | | | | | | |
| • VAE | 0.350 ±0.038 | 11 | **3.60** | 3.95 | 0.737 | **0.688** ±0.052 | **0.675** ±0.052 | 0.350 ±0.053 | 0.120 ±0.009 | — |
| • Cosmos | **0.406** ±0.039 | 9 | 3.57 | **4.01** | 0.722 | 0.637 ±0.054 | 0.500 ±0.056 | **0.438** ±0.055 | 0.132 ±0.011 | — |
| • V-JEPA 2.1 | 0.350 ±0.038 | 21 | 3.52 | 3.84 | **0.740** | 0.575 ±0.055 | 0.562 ±0.055 | 0.312 ±0.052 | 0.093 ±0.008 | — |
| • V-JEPA 2.1$_{96}$ | 0.388 ±0.039 | **8** | 3.44 | 3.80 | 0.737 | 0.688 ±0.052 | 0.550 ±0.056 | 0.287 ±0.051 | 0.106 ±0.008 | — |
| • Web-DINO | 0.325 ±0.037 | 14 | 3.39 | 3.69 | 0.737 | 0.588 ±0.055 | 0.500 ±0.056 | 0.325 ±0.052 | **0.087** ±0.007 | — |
| • Web-DINO$_{96}$ | 0.344 ±0.038 | 14 | 3.59 | 3.90 | 0.735 | 0.550 ±0.056 | 0.588 ±0.055 | 0.225 ±0.047 | 0.091 ±0.006 | — |
| • SigLIP 2 | 0.356 ±0.038 | 8 | 3.42 | 3.75 | 0.734 | 0.625 ±0.054 | 0.588 ±0.055 | 0.300 ±0.051 | 0.088 ±0.007 | — |
| • SigLIP 2$_{96}$ | 0.381 ±0.038 | 11 | 3.43 | 3.75 | 0.733 | 0.575 ±0.055 | 0.613 ±0.054 | 0.388 ±0.054 | 0.092 ±0.007 | — |

Table 11: **Per-instruction VLA task success for DiT-L encoders** (each cell: successes out of 8 trials). Columns: ● VAE, ● Cosmos, ● V-JEPA $2.1_{96}$, ● Web-DINO$_{96}$, ● SigLIP $2_{96}$. Thin gray rule separates reconstruction encoders from semantic encoders within each rater group. Instructions are ranked into four difficulty levels for a tabletop robotic arm: L1 basic single-step actions; L2 pick-and-place with relative/area positioning; L3 precise placement, specific orientation, or force control; L4 deformable-object manipulation and structural stacking. **Bold**: highest count per instruction and rater (zero-only rows not highlighted).

| | Qwen 3.6 | | | | | InternVL 3.5 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Instruction | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| *Level 1 — Basic single-step actions (open, close, sweep)* | | | | | | | | | | |
| close oven | 6 | 6 | 7 | **8** | 6 | **8** | **8** | **8** | **8** | **8** |
| open the drawer | **8** | **8** | **8** | **8** | **8** | **8** | **8** | **8** | **8** | **8** |
| pick up sponge and wipe plate | **8** | 7 | **8** | **8** | **8** | **8** | **8** | 7 | 5 | **8** |
| sweep into pile | 7 | 7 | 7 | **8** | 6 | **8** | **8** | 7 | 4 | 5 |
| *mean* | 7.2 | 7.0 | 7.5 | **8.0** | 7.0 | **8.0** | **8.0** | 7.5 | 6.2 | 7.2 |
| *Level 2 — Pick-and-place with relative or area-level positioning* | | | | | | | | | | |
| Move the can behind the blue fork | 2 | 3 | 2 | 6 | **7** | **7** | 4 | **7** | **7** | 6 |
| Move the red spoon to the left of the pot | **7** | 6 | 5 | 6 | **7** | 6 | **8** | 7 | **8** | **8** |
| close brown1fbox flap | 1 | **2** | **2** | 0 | **2** | 7 | 4 | **8** | 2 | 6 |
| moved the blue scrubber onto the lower right burner | 4 | 5 | 5 | **6** | 3 | 2 | 3 | **4** | **4** | 2 |
| pick up the green object above the drawer and place it on the table | 0 | 0 | 1 | **2** | 1 | 1 | 1 | 1 | **3** | 1 |
| place the silver pot in the middle of the table | 1 | **4** | **4** | 3 | 0 | **2** | 1 | **2** | 0 | 0 |
| put banana in pot or pan | 5 | **6** | 5 | 3 | 5 | **4** | **4** | 1 | 0 | 0 |
| *mean* | 2.9 | **3.7** | 3.4 | **3.7** | 3.6 | 4.1 | 3.6 | **4.3** | 3.4 | 3.3 |
| *Level 3 — Precise placement, specific orientation, or force control* | | | | | | | | | | |
| pick up blue towel from the grey thing and placed it to the right of the white basket | 4 | 5 | 5 | 5 | **8** | 6 | 5 | **7** | 6 | 6 |
| pour almonds in pot | 4 | 2 | 1 | 1 | **8** | 0 | 1 | 0 | 0 | **2** |
| put cucumber in cup | 4 | 3 | 4 | **7** | 3 | 1 | 1 | **3** | **3** | **3** |
| put the covering lid on top of the silver pot | 3 | **4** | 1 | 3 | 3 | 1 | 0 | 0 | **2** | 1 |
| turn lever vertical to front | **4** | 1 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| *mean* | 3.8 | 3.0 | 2.6 | 3.4 | **4.6** | 1.6 | 1.4 | 2.0 | 2.2 | **2.4** |
| *Level 4 — Deformable-object manipulation and structural stacking* | | | | | | | | | | |
| fold the cloth from the bottom to the top | 7 | **8** | 7 | **8** | 7 | 5 | **6** | 5 | 0 | 2 |
| move the red rectangle from one tower to another | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | **2** | 1 |
| put the rectangular block on top of the yellow and blue cubes | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| unfold the cloth from bottom right to top left | 6 | **8** | 7 | 6 | 5 | **8** | **8** | 7 | 6 | 5 |
| *mean* | 3.2 | **4.0** | 3.5 | 3.5 | 3.0 | **3.5** | **3.5** | 3.0 | 2.0 | 2.0 |

### D.3 Statistical Analyses

Table 12: **Uncertainty estimates for policy-facing metrics.** Cells show means with 95% bootstrap confidence intervals. VLA SR uses consensus VLM success; OOD SR pools distractor and instruction shifts; CEM is one-step controllability error. Family-level rows compare semantic encoders against reconstruction encoders. **Best** and runner-up are scoped per column.

| Encoder | VLA SR↑ | OOD SR↑ | CEM error↓ |
|---|---|---|---|
| • VAE | 0.169 [0.113, 0.231] | 0.303 [0.244, 0.366] | 0.111 [0.096, 0.129] |
| • VA-VAE | 0.175 [0.119, 0.237] | 0.225 [0.163, 0.294] | 0.097 [0.095, 0.120] |
| • Cosmos | 0.244 [0.181, 0.312] | 0.319 [0.250, 0.388] | 0.112 [0.096, 0.130] |
| • V-JEPA 2.1 | **0.344** [0.269, 0.419] | **0.487** [0.412, 0.569] | 0.084 [0.070, 0.100] |
| • Web-DINO | 0.212 [0.150, 0.275] | 0.388 [0.319, 0.456] | 0.090 [0.078, 0.103] |
| • SigLIP 2 | 0.325 [0.256, 0.400] | 0.381 [0.306, 0.456] | **0.082** [0.071, 0.094] |
| FAMILY-LEVEL SEMANTIC VS. RECONSTRUCTION TESTS | | | |
| Semantic − reconstruction | +0.098 [0.025, 0.177] | +0.136 [0.088, 0.184] | -0.0266 [-0.0412, -0.0122] |
| One-sided test | $p = 0.0129$ | $p < 5{\times}10^{-5}$ | $p = 0.00015$ |

**Uncertainty over policy-facing metrics.** The results show the same simple pattern across the policy-facing metrics: semantic latent spaces are better for task-relevant behavior than reconstruction latent spaces. For in-distribution VLA rollouts, semantic encoders exceed reconstruction encoders by 9.8 percentage points, with a 95% paired bootstrap interval of [2.5, 17.7] points and an exact one-sided sign-flip test of $p = 0.0129$ over the 20 shared task episodes. The OOD result is also positive: when pooling distractor and instruction shifts, semantic encoders exceed reconstruction encoders by 13.6 percentage points, with a 95% bootstrap interval of [8.8, 18.4] points and $p < 5{\times}10^{-5}$. For CEM action recovery, lower error is better; semantic encoders reduce one-step controllability error by 0.0266, with a 95% bootstrap interval of [0.0122, 0.0412] lower error and $p = 0.00015$. Thus, the semantic-family advantage is statistically supported for VLA success, OOD success, and CEM action recovery.

### D.4 Latent representation quality

Table 13: **Inverse Dynamics Model action-recovery** (Pearson $r$ averaged over action dimensions) for horizons $k$=1 and $k$=4. *Real* = on encoded GT latents (the encoder ceiling); *WM* = on world-model rollouts. **Best** and runner-up per column.

| | DiT-S | | | | DiT-B | | | | DiT-L | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $k$=1 | | $k$=4 | | $k$=1 | | $k$=4 | | $k$=1 | | $k$=4 | |
| Encoder | Real↑ | WM↑ | Real↑ | WM↑ | Real↑ | WM↑ | Real↑ | WM↑ | Real↑ | WM↑ | Real↑ | WM↑ |
| • VAE | 0.507 | 0.476 | 0.478 | 0.464 | 0.507 | 0.495 | 0.478 | 0.470 | 0.507 | 0.510 | 0.478 | 0.483 |
| • VA-VAE | 0.549 | 0.545 | 0.744 | 0.719 | — | — | — | — | — | — | — | — |
| • Cosmos | 0.626 | 0.581 | 0.673 | 0.651 | — | — | — | — | 0.626 | 0.617 | 0.673 | 0.671 |
| • V-JEPA 2.1 | **0.829** | **0.781** | **0.865** | **0.840** | **0.829** | **0.779** | **0.865** | **0.834** | **0.829** | **0.797** | **0.865** | **0.848** |
| • Web-DINO | 0.820 | 0.729 | 0.845 | 0.794 | 0.820 | 0.778 | 0.845 | 0.824 | 0.820 | 0.705 | 0.845 | 0.785 |
| • SigLIP 2 | 0.772 | 0.697 | 0.793 | 0.757 | — | — | — | — | 0.772 | 0.705 | 0.793 | 0.762 |

Table 14: **Trajectory success-probe accuracy across DiT sizes.** *Enc. Acc/AUC* is computed on encoded ground-truth latents (the probe ceiling); per-DiT columns are accuracy on world-model rollouts and the absolute *Drop* from the encoder ceiling (lower is better). **Best** and runner-up per column. Dashed rule separates VAE-like and SSL encoders.

| | | | DiT-S | | DiT-B | | DiT-L | |
|---|---|---|---|---|---|---|---|---|
| Encoder | Enc. Acc | Enc. AUC | Acc | Drop↓ | Acc | Drop↓ | Acc | Drop↓ |
| • VAE | 0.835 | 0.917 | 0.716 | 0.119 | 0.716 | 0.119 | 0.685 | 0.150 |
| • VA-VAE | 0.868 | 0.938 | 0.744 | 0.124 | — | — | — | — |
| • Cosmos | 0.851 | 0.925 | 0.723 | 0.128 | — | — | 0.732 | 0.119 |
| • V-JEPA 2.1 | 0.905 | 0.963 | 0.789 | 0.116 | **0.791** | **0.114** | 0.796 | 0.109 |
| • Web-DINO | **0.906** | **0.963** | 0.788 | 0.118 | 0.789 | 0.117 | 0.797 | 0.109 |
| • SigLIP 2 | 0.903 | 0.961 | **0.823** | **0.080** | — | — | **0.827** | **0.076** |

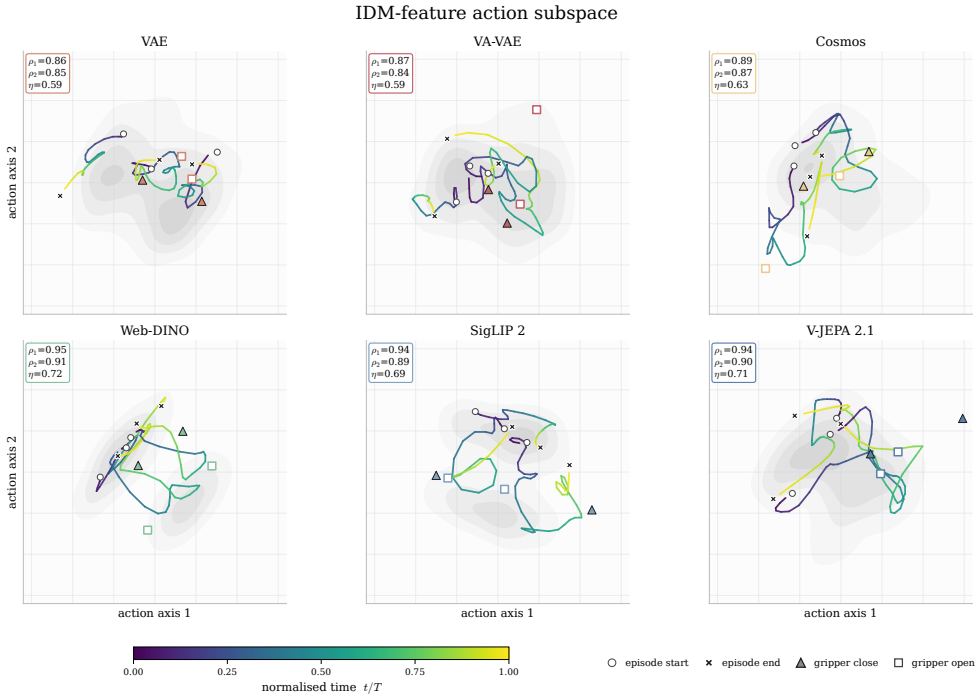



Figure 10: **Action trajectories induced by encoder spaces:** episode rollouts projected onto the top-2 canonical-correlation directions between IDM features and ground-truth actions. $(\rho_1, \rho_2)$ are the leading canonical correlations, $\eta$ summarizes the aggregate action alignment. Colored curves are episodes.

## D.5 Multi-view transfer learning

Table 15: **DiT-S single-view vs multi-view.** Each cell for PSNR and LPIPS shows the WM value with the gap to its encoder's reconstruction ceiling in parentheses (smaller = closer to ceiling). **Best** and runner-up per column across all rows; the WM value and gap are highlighted independently. The two adapter pairs (V-JEPA $2_{96}$, Web-DINO$_{96}$) only have multi-view data for CEM. **Best** within each column.

| | Reconstruction fidelity | | Generative quality | | Controllability |
|---|---|---|---|---|---|
| Encoder | PSNR ↑ | LPIPS ↓ | FID ↓ | FVD ↓ | CEM L2 ↓ |
| • VAE | 17.43 (16.65) | 0.218 (0.207) | 17.43 | 6.8 | 0.111 |
| • VAE (multi) | 16.67 (18.90) | 0.234 (0.226) | 22.03 | 12.9 | **0.047** |
| • Cosmos | 16.97 (10.06) | 0.245 (0.197) | 16.95 | 8.2 | 0.112 |
| • Cosmos (multi) | 16.07 (12.02) | 0.266 (0.223) | 27.65 | 13.8 | 0.050 |
| • V-JEPA 2.1 | **18.10** (11.09) | **0.176** (**0.141**) | 6.77 | 5.5 | 0.084 |
| • V-JEPA 2.1 (multi) | 17.50 (11.10) | 0.186 (0.145) | 9.18 | 6.2 | 0.056 |
| • Web-DINO | 17.42 (10.87) | 0.199 (0.160) | 7.63 | 6.7 | 0.090 |
| • Web-DINO (multi) | 17.43 (**9.77**) | 0.191 (0.141) | 10.12 | 7.3 | 0.052 |

### D.6 Effect of adapter dimension

We observe that adapter dimension has a non-monotonic sweet spot. Table 16 shows that the adapter bottleneck dimension has a non-monotonic effect on performance. For Web-DINO with DiT-S, the intermediate $d_{96}$ setting gives the best overall trade-off, achieving the highest VLA success rate and the best LPIPS, FID, and FVD. Smaller bottlenecks such as $d_{16}$ remain competitive for policy performance but lose visual quality, while using the full $D_{1024}$ encoder output is worse than the compact $d_{96}$ adapter.

Table 16: **Adapter dim.** $d$ **ablation** for Web-DINO DiT-S. **Best** and runner-up highlighted per row.

| | Web-DINO (DiT-S) latent dim | | |
|---|---|---|---|
| Metric | $d_{16}$ | $d_{96}$ | $D_{1024}$ |
| VLA SR↑ | 0.256 | **0.269** | 0.181 |
| SSIM↑ | 0.711 | **0.728** | 0.722 |
| LPIPS↓ | 0.196 | **0.181** | 0.199 |
| FID↓ | 8.37 | **6.00** | 7.63 |
| FVD↓ | 7.65 | **5.51** | 6.66 |

## E Additional Rollouts

We provide additional rollouts alongside the key observations for Open-VLA success rate comparison (Fig. 11), plain pixel rollouts for comparing differences between standard model outputs (Fig. 12) and hallucinated model outputs (Fig. 13), rollouts under OOD distractor objects as well as under OOD instructions for all models across diverse episodes (Fig. 14, 16) as well as on the same episode (Fig. 15, 17). We also provide sample rollout videos for analyses with the supplementary files.

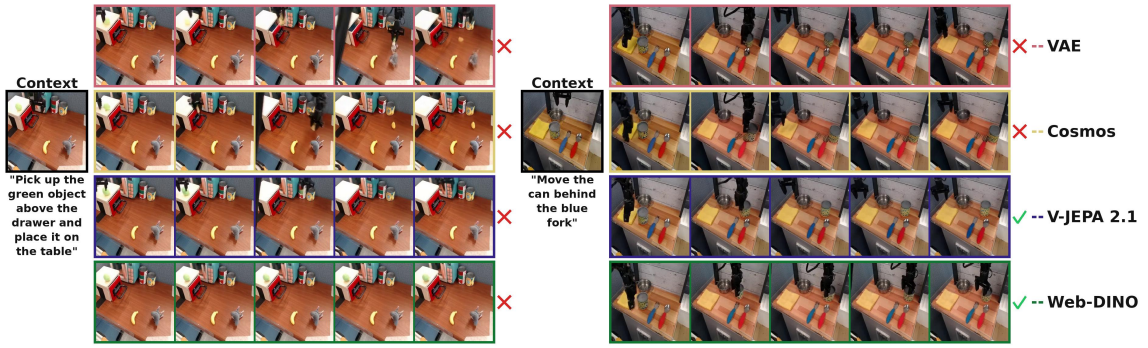



Figure 11: **Open-VLA success rate comparison on two random episodes:** four frames are sampled at even intervals. ✓ and ✗ show trajectories marked as success and failure by InternVL 3.5 VLM.

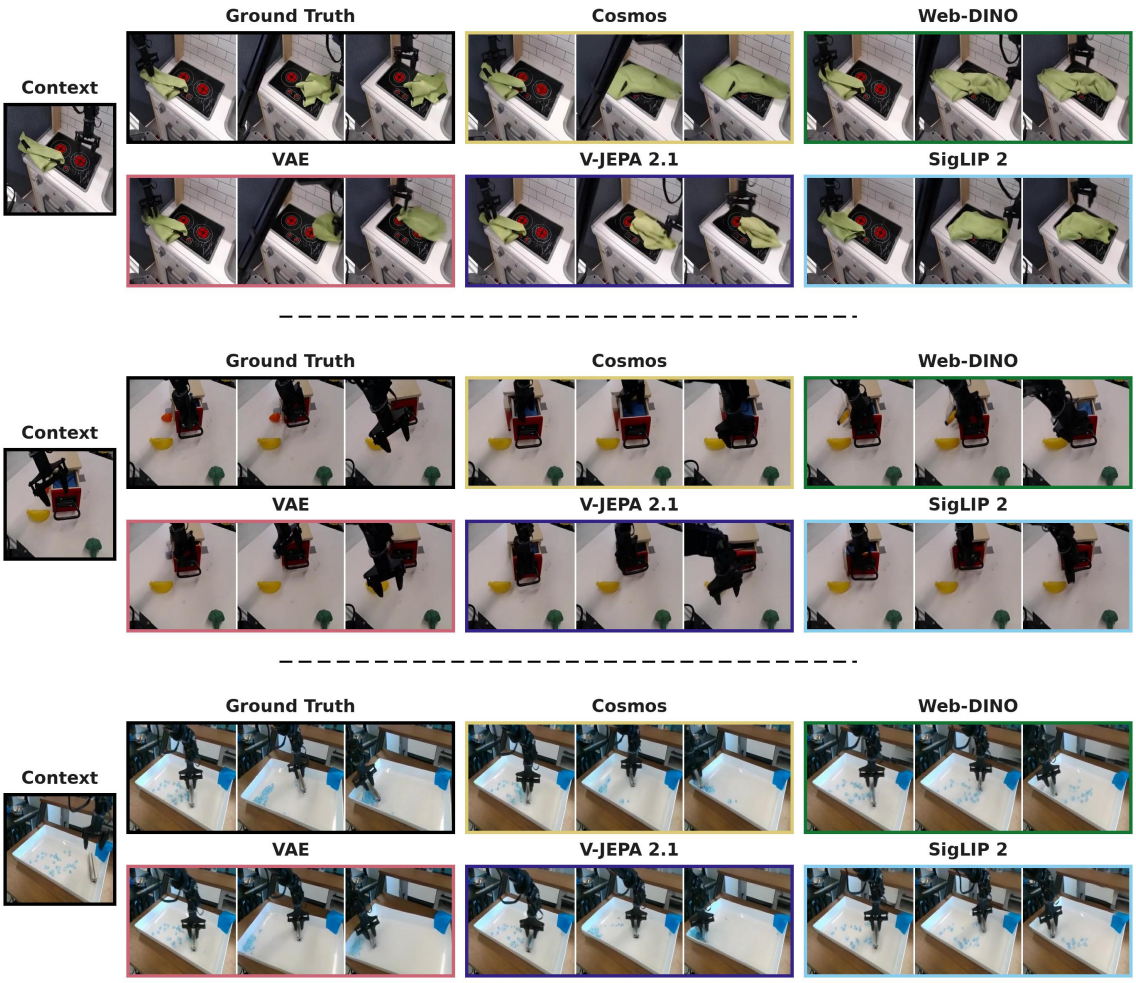



Figure 12: **Pixel rollout comparison across models on diverse episodes:** the first frame is fed as context and the rest 3 frames are sampled at even intervals from the generated world model rollout.

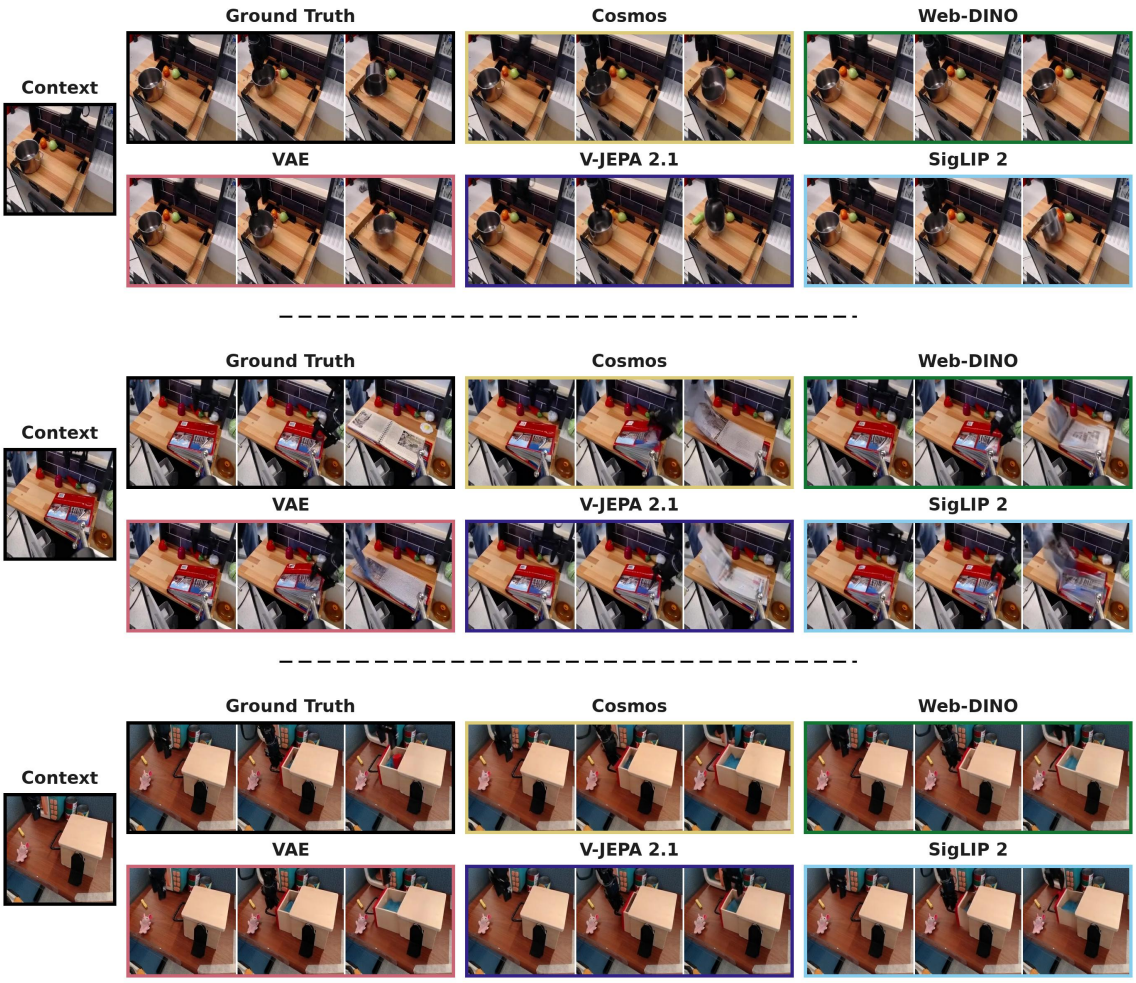



Figure 13: **Hallucinated pixel rollout comparison across models on diverse episodes:** the first frame is fed as context and the rest 3 frames are sampled at even intervals from the generated world model rollout. **Top:** flipping the pot consistently causes distortions for all models; **Middle:** turning the book pages causes the models to only partially follow the motion with the book/page appearances becoming smeared and inconsistent, the page edges and cover boundaries drifting; **Bottom:** while all models predict the appearance of an opening drawer, some clearly under-predict the opening (e.g. VJEPA 2.1) while others show unstable drawer boundary and front panel (e.g. Cosmos).

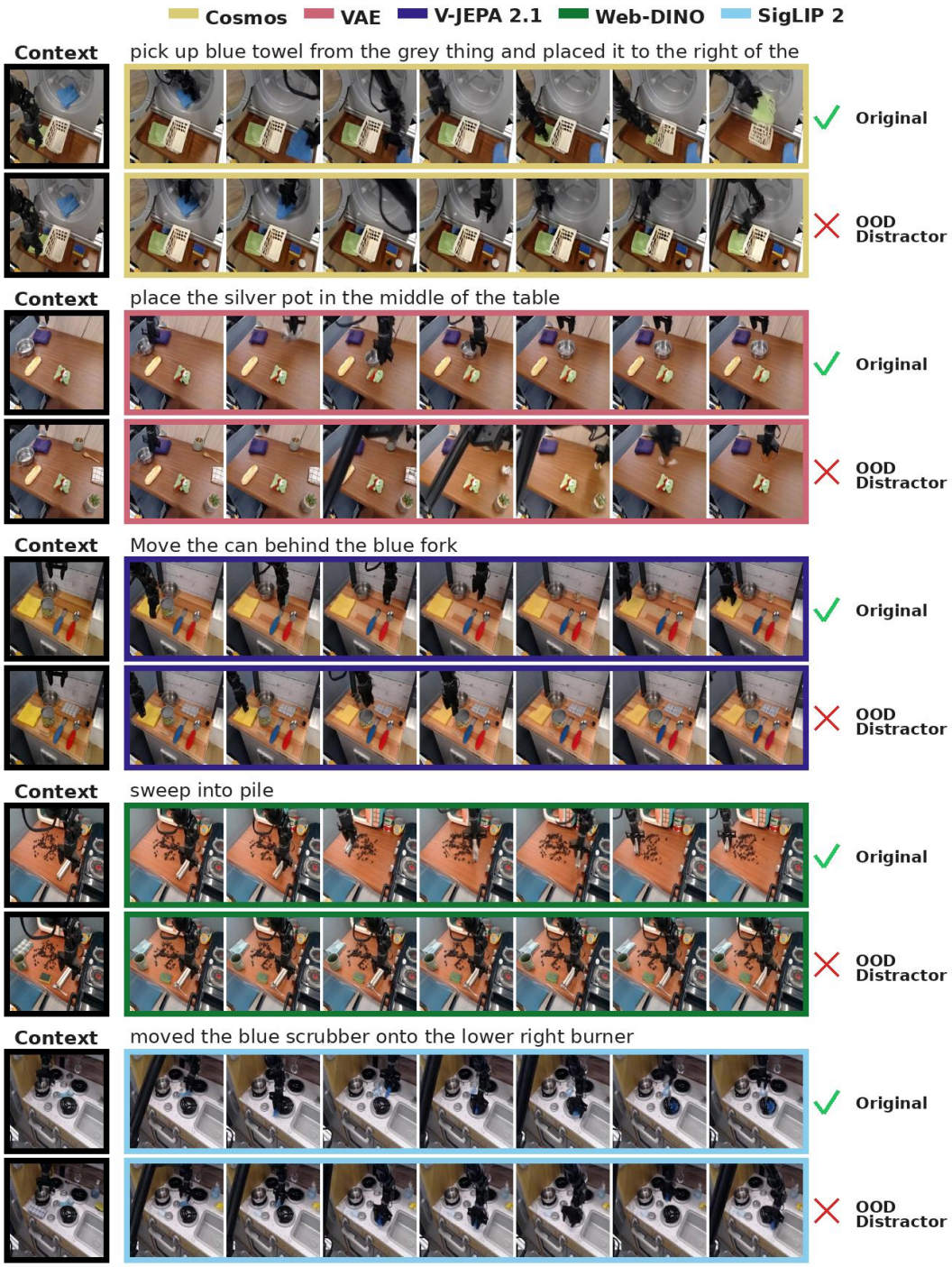



Figure 14: **OOD Distractor comparison showing failure episodes per model:** OOD objects break task-object binding and action-conditioned state tracking across all models. ✓ and ✗ show trajectories marked as success and failure by InternVL 3.5. In their respective trajectories: Cosmos generates less stable towel/object state; VAE fails at task-relevant placement of the silver pot; V-JEPA 2.1 loses stable binding between the can, the blue fork, and the instruction, with the can failing to end up reliably behind the fork; Web-DINO fails to maintain the pile-forming interaction; SigLIP 2 keeps the stove layout recognizable, but it does not preserve the precise relation between the scrubber and the target-burner.


Figure 15: **OOD Distractor comparison for the same episode:** OOD distractor competes with the target objects and exposes whether a model can keep the target objects bound to the instruction. ✓ and ✗ show trajectories marked as success and failure by InternVL 3.5. Here, irrespective of the task success, the added object visually changes the predicted interaction for all models: the robot/can motion becomes less task-directed, the cans position is less consistently moved behind the blue fork, and the models appear to let the distractor alter the scene dynamics.

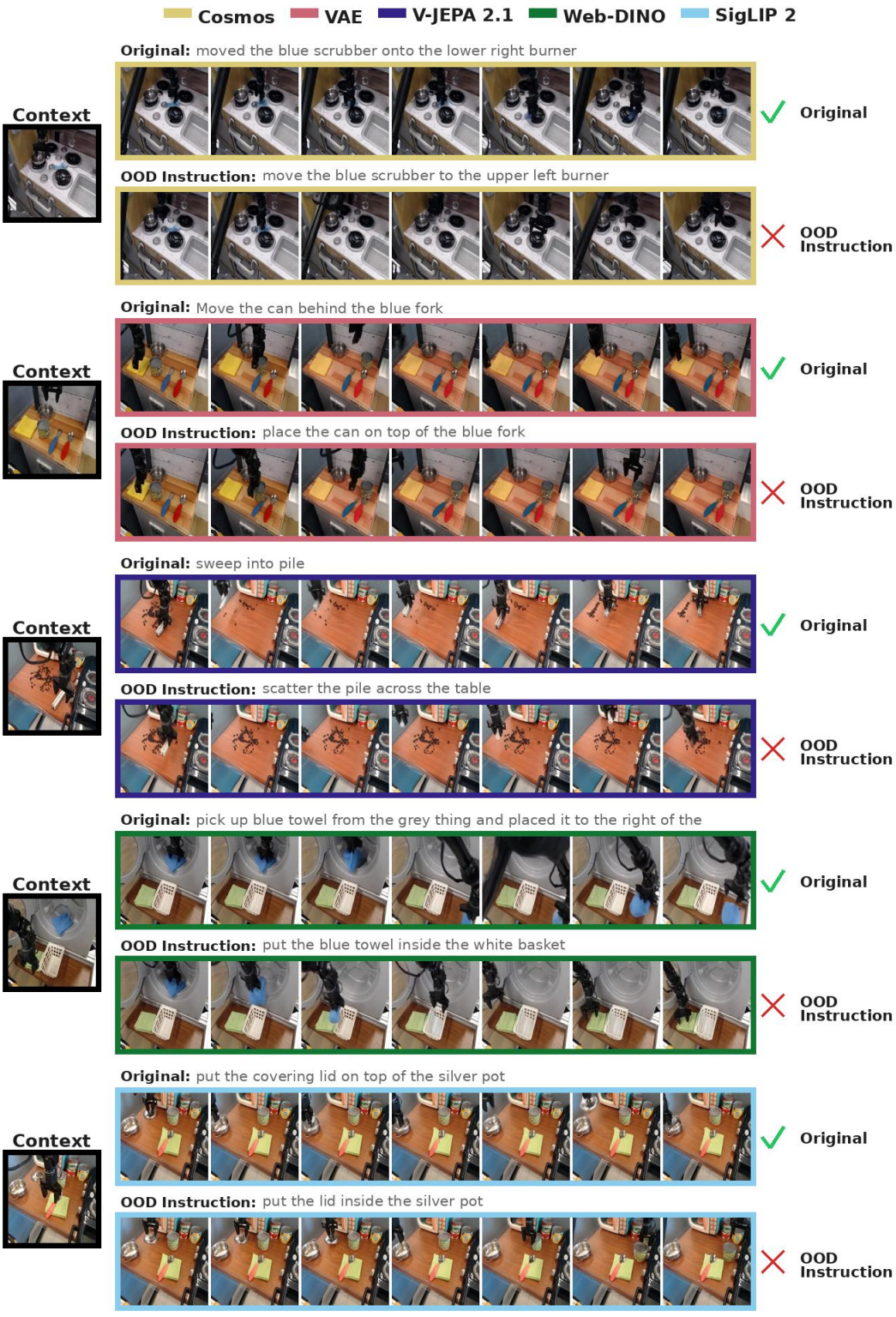



Figure 16: **OOD Instruction comparison showing failure cases per model:** for each model, the same initial context is rolled out with the original instruction, which succeeds, and then with a OOD instruction, which fails. ✓ and ✗ show trajectories marked as success and failure by InternVL 3.5. Model-specific OOD instruction trajectories show: Cosmos rollout still moves the blue scrubber around the stove, but does not reliably bind it to the new target burner; VAE preserves the table scene, but fails to understand the spatial relation "on top of"; VJEPA 2.1 rollout continues to look like sweeping/piling behavior rather than reversing the task into scattering; Web-DINO keeps the towel manipulation plausible, but misses the new container-based goal; SigLIP 2 rollout shows the lid disappear off the frame.


Figure 17: **OOD Instruction comparison for the same episode:** most models exhibit a common hallucination where the original object dynamics or defaults to a familiar action pattern instead of updating the final state to match the new instruction. ✓ and ✗ show trajectories marked as success and failure by InternVL 3.5. Both Cosmos and VAE maintain the cloth in a partially folded/creased state instead of flattening it. Semantic encoders more clearly capture the semantic difference between folding and unfolding with VJEPA 2.1 most clearly producing a flatter cloth for the OOD instruction. Web-DINO spreads the cloth, but with some shape distortion and robot occlusion while for SigLIP 2, the cloth shape becomes rounded, suggesting some geometry hallucination despite correct task-level outcome.