VIDEO WORLD MODELS NEED PREDICTABLE LATENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Video world models trained on large-scale robot data promise a safe and scalable approach for learning and evaluating policies that act in the physical world. However, they still struggle to generate realistic long-horizon rollouts, generalize out of distribution, and they remain computationally demanding at inference time. The dominant approach, making predictions in the latent space of a reconstruction-trained autoencoder, does not incentivize a latent geometry that facilitates dynamics modeling. For this reason, we investigate whether encouraging a meaningful temporal organization in latent space can benefit downstream world model performance, using a series of controlled experiments. We find that temporal representation learning objectives can improve downstream performance of the world model at matched reconstruction quality. However, the best-performing objectives need over 20x more autoencoder updates to reach the same reconstruction performance. To address this issue, we characterize the trade-off between information retention and useful geometric organization in latent space training, and we propose a simple modification to the architecture of the decoder that improves it. We find that pairing this architecture with temporal objectives for autoencoder finetuning improves downstream performance of the world model at matched training budget. Compared to a reconstruction baseline, the gains include up to 5x faster inference, 8x faster training, 24% MSE decrease out of distribution and 18% MSE decrease in long-horizon extrapolation. This also translates into better agreement of the world model with real-world policy evaluation, with up to a 0.46 increase in Pearson correlation using 2-step inference. Common to the best temporal objectives is a large increase in one-step predictability of the latents, lower intrinsic dimensionality, and weaker high frequency components. Overall, our study indicates that temporal predictability is an important design criterion for the latent space of video world models, and provides concrete guidance on how to incorporate it into autoencoder training.

1 INTRODUCTION

Learning from experience with high-capacity function approximators has led to unprecedented capabilities in domains like games (Silver et al., 2018), mathematics (Hubert et al., 2026), and programming (OpenAI, 2025). These successes have been enabled by two key ingredients: a verifiable reward signal to drive policy improvement, and a cheap, safe, parallelizable way to interact with the environment and establish a tight feedback loop. Extending these results to domains like robotics, autonomous driving, and scientific research is challenging because a reward function can be difficult to specify, and interaction with the environment is often costly and unsafe. While these difficulties have been overcome in specific applications, like drone racing (Kaufmann et al., 2023) and robotic table tennis (Dürr et al., 2026), this has required substantial engineering effort, especially in building an accurate enough simulator for policy training and sim-to-real transfer.

Recently, video world models (VWMs) have emerged as a promising approach to enable learning from experience in non-verifiable, costly, and unsafe domains, including the physical world. By learning a photorealistic action-conditioned simulator of the environment from video data, these models allow cheap, safe, and parallelizable interaction with the environment (Hafner et al., 2025b; Sharma et al., 2026). Therefore, paired with a general-purpose vision-language model (VLM) acting as a reward model, VWMs enable policy evaluation (Quevedo et al., 2026), policy training via RL (Sharma et al., 2026), and planning via imagined rollouts (Du et al., 2024). However, despite

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

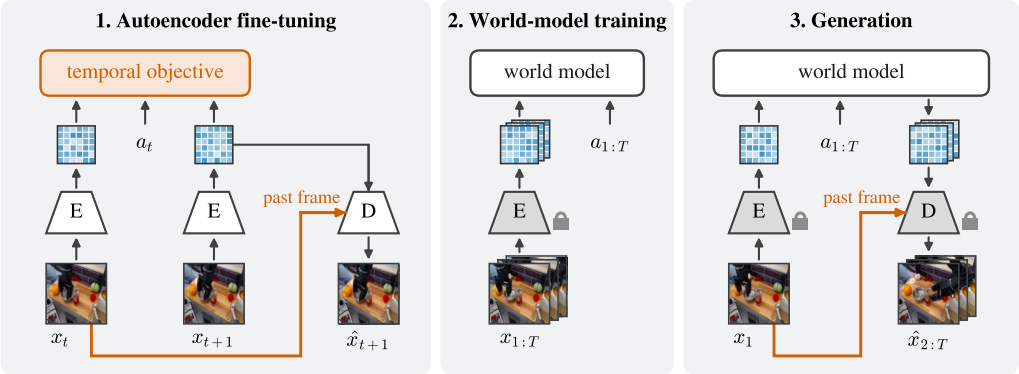


Figure 1: **Method overview.** (1) A pretrained autoencoder is fine-tuned on pairs of nearby frames with a reconstruction loss and a temporal objective on the two latents and the action a_t between them; the decoder D also receives the past frame x_t. (2) The temporal objective is dropped, and an action-conditioned diffusion world model is trained on the latents of the frozen encoder E. (3) From a context frame x_1 and the actions, the world model generates latents, which D decodes with x_1 as the past frame of every generated frame.
Highlighted in orange: the two components we add to a standard latent video world model; without them, the pipeline is our baseline.

their promise, VWMs still struggle to generate accurate long-horizon rollouts (Guo et al., 2026b; Bardhan et al., 2026; Huang et al., 2026a), especially out of distribution (Tseng et al., 2025; Zhang et al., 2026a; Kang et al., 2025; Chen et al., 2026), and they remain computationally demanding at inference time (Quevedo et al., 2026; Gemini Robotics Team, 2025; Jeon et al., 2026; Lu et al., 2026).

Following common practice in visual generative modelling (Rombach et al., 2022; Blattmann et al., 2023; Peebles & Xie, 2023), most existing VWMs make predictions in the latent space of an autoencoder learned through reconstruction (Bruce et al., 2024; Decart et al., 2024; NVIDIA, 2025; Quevedo et al., 2026; Huang et al., 2026b). However, while it encourages retaining fine-grained information about the observations, a reconstruction objective does not directly incentivize latents whose dynamics is easy to model (Yao et al., 2025; Kouzelis et al., 2025; Skorokhodov et al., 2025). A different line of work considers world models that learn to predict in latent space without ever reconstructing observations (LeCun, 2022; Hansen et al., 2024; Zhou et al., 2025a; Sobal et al., 2025; Assran et al., 2025; Maes et al., 2026). This grants them freedom to discard information that is hard to predict (Van Assel et al., 2025), and thus to use compact latents that make them lightweight and efficient (Maes et al., 2026). However, that same freedom creates the risk of representation collapse, requiring dedicated and sometimes poorly understood mechanisms to mitigate it (Grill et al., 2020; Bardes et al., 2022; Balestriero & LeCun, 2025; Maes et al., 2026). Collapse, even partial, is at odds with the idea of leveraging powerful general-purpose VLMs as reward models for reinforcement learning, since the VLM judges task success based on frames decoded from the rollouts (Quevedo et al., 2026; Sharma et al., 2026), and VLM judges fail when task-relevant detail is not visible (Pendharkar, 2026; Navasardyan et al., 2026).

Motivated by these complementary strengths, in this work we investigate whether it is possible to get the best of both worlds: a latent space that is easily predictable without sacrificing the ability to accurately reconstruct the observations. To answer this question, we consider several families of representation learning objectives that use action-labelled videos to give the latent space temporal structure: latent future prediction, pixel future prediction, contrastive future prediction, and inverse dynamics (Sermanet et al., 2018; van den Oord et al., 2018; Nachum & Yang, 2021; Hansen et al., 2024). We combine these objectives with a reconstruction loss, and we carry out controlled experiments where we fine-tune a general-purpose visual autoencoder (Black Forest Labs, 2024), controlling either for reconstruction error or for training budget (Figure 1). We then measure the quality of the learned representations by evaluating the downstream performance of latent VWMs trained on robotic manipulation data (Walke et al., 2023), measuring prediction error in and out of distribution (Zhou et al., 2025b), as well as policy evaluation ability (Quevedo et al., 2026).

Our main contributions are the following.

- **Temporal structure is useful, but it fights information retention.** We find that including temporal objectives in autoencoder fine-tuning leads to improved world model performance, controlling for the reconstruction error of the autoencoder. The most effective objectives are those that encourage predictability in latent space directly. However, the training budget required to reach a given reconstruction loss target becomes much larger compared to pure reconstruction training, especially for the best performing objectives; this is associated with a lower intrinsic dimensionality of the learned latents.
- **Past-conditioned decoding improves the information-structure frontier.** We characterize the trade-off between the information a latent retains and the useful structure it acquires by varying the relative importance of reconstruction and the temporal objective. To afford more predictable latents at a fixed reconstruction error, we propose to modify the decoder architecture so that its outputs are conditioned on a past observation from the environment. We show that this simple measure is effective at improving the characterized trade-off.
- **A practical recipe to improve training of world models:** combining the past-conditioned decoder with temporal objectives like latent prediction or inverse dynamics. We show that, compared to the standard practice at matched training budget, world models trained on our latents obtain up to 8x faster training in LPIPS and SSIM, 5x fewer sampling steps at matched generation quality, a 15-24% improvement in pixel and perceptual error out of distribution and extrapolating to long horizon (Figure 10), and large gains in the ability to evaluate policies, with up to a 0.46 increase in Pearson correlation with real world evals using 2-step inference. Both the decoder and the objectives account for part of these gains. Furthermore, we observe that both objectives increase the predictability of the latents, with a 33-80% reduction in one-step forecast error, despite inverse dynamics not explicitly training for this.

2 Related work

Video world models. Video world models generate future frames conditioned on past frames and actions, autoregressively over discrete visual tokens (Bruce et al., 2024; Wu et al., 2024) or, increasingly, with diffusion (Alonso et al., 2024; Zhu et al., 2025; NVIDIA, 2025; Quevedo et al., 2026). Apart from a few pixel-space models (Yang et al., 2024; Alonso et al., 2024), they predict in the latent space of an autoencoder trained to reconstruct, with pixel, perceptual and adversarial losses, and sometimes a small KL penalty (Rombach et al., 2022; Yu et al., 2024; Yang et al., 2025; Kong et al., 2024; Valevski et al., 2025; NVIDIA, 2025; Wan Team, 2025; Guo et al., 2026b). While some recent works have adopted spatiotemporal autoencoders (Yu et al., 2024; Kong et al., 2024; NVIDIA, 2025; Wan Team, 2025), time mostly serves as an additional compression axis, and sometimes enters the definition of the reconstruction objective in pixel space, like with clip-level discriminators (Yu et al., 2024; Yang et al., 2025; Wan Team, 2025; Russell et al., 2025), wavelet sub-bands (HaCohen et al., 2024; Li et al., 2025), or an optical-flow loss between input and reconstruction (NVIDIA, 2025). In contrast, we focus on temporal objectives that affect the relationship between past and future representations, and we find that their gains are largest when they act on the latents directly (Section 5.1).

Reconstruction-free world models. Another line of work learns predictable latents without ever reconstructing observations: the encoder is trained jointly with a latent dynamics model to predict future latents, rewards, or values (Hansen et al., 2024; Sobal et al., 2025; Maes et al., 2026; Okada & Taniguchi, 2021; Nguyen et al., 2021; Deng et al., 2022). This lets the latents discard content that is hard to predict or irrelevant to the task (Nguyen et al., 2021; Balestriero & LeCun, 2024; Van Assel et al., 2025), but also makes these models susceptible to total or partial collapse, whereby latent prediction can favour slow or easily predictable features and discard task-relevant ones (Sobal et al., 2022; Littwin et al., 2024; Pendharkar, 2026; Karmann et al., 2026). Consequently, several of these works report that decoders trained on the latents a posteriori lose content: background texture (Okada & Taniguchi, 2021; Nguyen et al., 2021), the orientation of the end effector in long predicted rollouts (Maes et al., 2026), or the foreground configuration when the background is static (Karmann et al., 2026). This limits their ability to leverage generalist VLMs as a judge for policy training, policy evaluation, and planning with the world model (Tseng et al., 2025; Jeon et al., 2026; Sharma et al., 2026; Navasardyan et al., 2026). In contrast, we keep a reconstruction objective, which a VLM judge needs, and add temporal objectives for predictability.

Latent space design for generative models. In image generation, autoencoders that reconstruct better do not necessarily support better generative models (Yao et al., 2025; 2026). Proposed remedies regularize the latent space (Kouzelis et al., 2025; Skorokhodov et al., 2025), align it with the features of a pretrained semantic encoder (Yao et al., 2025), or use those features directly as the latent space (Zheng et al., 2026). Video generation and VWMs have also adopted the semantic remedies, with frozen image or video encoders (Guo et al., 2026a; Xie et al., 2026; Zhang et al., 2026b; Hu et al., 2026) or alignment losses on the video tokenizer (Hu et al., 2023; Russell et al., 2025; FSVideo Team, 2026; Wang et al., 2026). Closest to our setting, Nilaksh et al. (2026) compare frozen, off-the-shelf reconstruction and semantic encoders as latent spaces for action-conditioned VWMs on Bridge V2. In the semantic approaches the structure of the latent space is inherited from a pretrained encoder, so the effect of its training objective is not separated from that of its data, architecture, and scale. We instead keep the autoencoder fixed and vary only its objective. Furthermore, we study objectives that relate past and future latents rather than spatial regularization or alignment with a pretrained encoder; our own experiments fine-tuning a DINOv2 encoder indicate that the two are complementary (Appendix H.9).

Temporal objectives on the autoencoder. In model-based reinforcement learning, the encoder is trained jointly with a latent dynamics model, which makes its latents predictable (Hafner et al., 2018; 2025a). Video world models, Dreamer 4 included (Hafner et al., 2025b), instead train their autoencoder separately, usually for reconstruction alone, and then freeze it. A few works add a temporal term to the autoencoder objective for video generation: future-frame prediction by the decoder (Zhao et al., 2026), slowness and uniformity losses for VWMs in deterministic synthetic environments (Xia et al., 2026), a motion-consistency loss on the robot-arm latents of a disentangled video autoencoder for robot VWMs (Luo et al., 2026), and a local spatiotemporal correlation loss for text-to-video generation (Liu et al., 2026). Zhao et al. (2026) observe that their objective costs reconstruction and partly recover it by fine-tuning the decoder; however, to our knowledge, none of these works compares families of temporal objectives, none uses actions to shape the representations, and none removes the reconstruction cost: with a past-conditioned decoder, our temporally structured latents can reconstruct better than those of a standard autoencoder trained for reconstruction alone. Finally, Yeom et al. (2026) consider objectives similar to ours, but their goal is finding features from which an inverse-dynamics probe recovers the executed action, in simulation, while we explore latents that can improve a VWM on real-world data.

3 Background

Latent video world models. We consider trajectories of frames $x_{1:T}$ and actions $a_{1:T}$, where a_t is the action executed at frame x_t; a VWM models the next frame given the past, $p(x_{t+1} \mid x_{1:t}, a_{1:t})$. In the latent VWMs we consider, an encoder E maps each frame independently to a latent $z_t = E(x_t)$, and a decoder D maps it back to a frame $\hat{x}_t = D(z_t)$. The autoencoder is typically trained as in latent diffusion (Rombach et al., 2022), with a reconstruction loss $\mathcal{L}_{\text{rec}}(x_t, \hat{x}_t)$ that combines pixel, perceptual (Zhang et al., 2018), and adversarial terms, and a lightly weighted KL penalty (Kingma & Welling, 2014). It is then frozen, and the world model $p_\theta(z_{t+1} \mid z_{1:t}, a_{1:t})$ is learned on its latents.

Temporal objectives for representation learning. In latent future prediction, a predictor maps the current latent, and the action where one is available, to a prediction of a future latent, which is regressed onto the encoder's output for that frame (Bardes et al., 2024; Hansen et al., 2024; Sobal et al., 2025). In pixel future prediction, the prediction is decoded and compared with the future frame in pixel space (Oh et al., 2015; Finn et al., 2016). In contrastive future prediction, the prediction is trained with the InfoNCE loss (van den Oord et al., 2018) to identify the future frame among negatives, drawn from other trajectories (Nachum & Yang, 2021) or from other frames of the same sequence (van den Oord et al., 2018; Sermanet et al., 2018). In inverse dynamics, a head predicts the action from the latents of two consecutive frames (Agrawal et al., 2016; Pathak et al., 2017).

Diffusion-forcing world model. Diffusion forcing (Chen et al., 2024) trains a diffusion model over sequences in which every frame carries its own noise level: a network receives the noisy latents, their noise levels, and the actions, and learns to denoise every frame (Appendix D). When the network is causal in time, it learns to denoise a frame given a past at any level of corruption, including none. Generation is then autoregressive: given the clean context and the frames generated so far, each new frame is denoised from Gaussian noise in a few steps of the DDIM sampler (Song et al., 2021). We

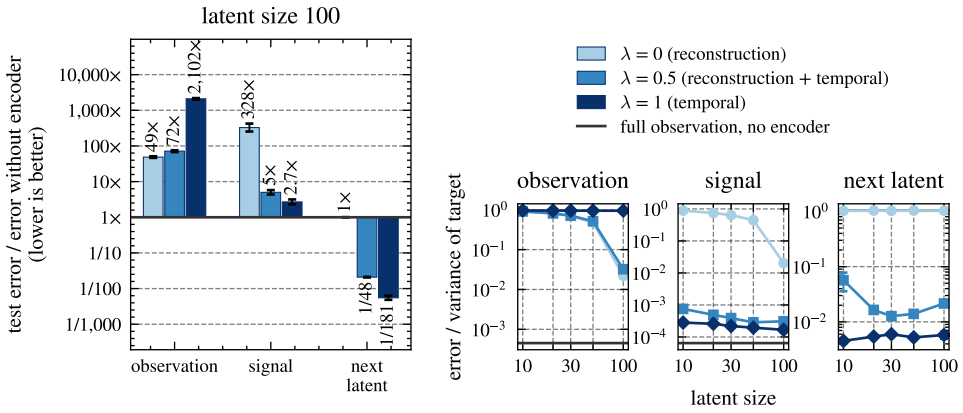


Figure 2: **Best of both worlds in the synthetic setting.** Errors of the observation, of its predictable part (signal) and of the next latent, predicted from latents learned with reconstruction ($\lambda = 0$), temporal contrastive learning ($\lambda = 1$) or both ($\lambda = 0.5$); mean and standard error over 3 seeds. **Left**: fixed latent dimension; errors divided by those of the full observation. **Right**: against the latent dimension; errors divided by the variance of the target.
Reconstruction makes the next latent hard to predict and needs a large latent to recover the signal; contrastive learning loses the observation; their combination avoids both failures.

use the space-time factorized diffusion transformer (DiT) (Peebles & Xie, 2023) of Quevedo et al. (2026), which alternates bidirectional attention within a frame with causal attention across frames (Appendix C).

4 Experiments with synthetic data

We first ask whether a latent can retain enough information to reconstruct the observations while being easy to predict.

Synthetic environment. We consider a synthetic environment where observations $x_t = (s_t, \xi_t) \in \mathbb{R}^{102}$ are given by the concatenation of a 2-dimensional vector s_t, the *signal*, evolving deterministically as $s_{t+1} = f(s_t, a_t)$, and a 100-dimensional vector ξ_t of *distractors*, sampled independently at every time step; every coordinate has unit variance.

Experimental design. An encoder E learns to map the observation to a latent $z_t = E(x_t) \in \mathbb{R}^d$ by optimizing the convex combination of a reconstruction objective (Kingma & Welling, 2014) and a temporal contrastive objective (van den Oord et al., 2018; Nachum & Yang, 2021):

$$\mathcal{L} = (1 - \lambda)\, \mathcal{L}_{\text{rec}} + \lambda\, \mathcal{L}_{\text{temp}} \tag{1}$$

where $\lambda \in [0, 1]$ weighs the two terms. On top of the frozen encoder, we train three modules: a dynamics model that predicts the next latent from the current latent and action, a probe that reads the signal from the latent, and a decoder that reconstructs the observation. On held-out data, we measure the errors of the reconstruction, of the signal, and of the next latent. Each error is divided by the variance of its target, so that predicting the mean gives an error of 1. As a reference, the same modules are trained directly on the observation, without an encoder. In Figure 2, we compare $\lambda = 0$, 0.5 and 1 at latent sizes d from 10 to 100; the data, networks, training, metrics and the full grid of weights are in Appendix A.

The combined objective achieves the best of both worlds. Reconstruction favours the leading principal components of the observation (Balestriero & LeCun, 2024); since every coordinate has unit variance, it recovers the signal only when the latent is nearly as large as the observation (signal error 0.89 at $d = 10$, 0.02 at $d = 100$), and its latent is as hard to predict as the observation (next-latent error about 1 at every size). The contrastive objective needs only the signal, which alone predicts

Table 1: **Temporal objectives at equal reconstruction.** Each autoencoder is fine-tuned until its reconstruction error reaches a target value, and the same world model is trained on its latents. First row: the baseline (reconstruction alone), mean and standard error over 3 seeds (MSE $\times 10^3$). Other rows: relative change (%) against the baseline with the same seed, mean over seeds, and number of seeds that improve; green: improved, orange: worsened; bold: best in each column. Updates: thousands of autoencoder updates to reach the reconstruction target; Dim.: principal components for 95% of the latent variance; mean and range over seeds.
Latent and contrastive prediction improve most metrics, but need far more updates to reach the same reconstruction. They also give latents with lower intrinsic dimension.

	Autoencoder		In distribution			Out of distribution			2 sampling steps			Long rollout, steps 51–59		
	Updates	Dim.	MSE	LPIPS	SSIM	MSE	LPIPS	SSIM	MSE	LPIPS	SSIM	MSE	LPIPS	SSIM
Baseline (reconstruction)	4.4	388 387–390	11.65 ±0.20	0.164 ±0.001	0.801 ±0.002	33.87 ±1.28	0.308 ±0.008	0.675 ±0.009	15.95 ±1.03	0.214 ±0.004	0.770 ±0.005	33.26 ±1.84	0.287 ±0.008	0.691 ±0.010
Latent prediction	156.5 146–171	177 162–185	−3.0 3/3	**−7.1** 3/3	**+1.5** 3/3	**−13.8** 3/3	**−3.1** 2/3	**+1.9** 2/3	**−21.9** 3/3	**−25.2** 3/3	**+5.0** 3/3	**−9.1** 3/3	**−8.4** 3/3	**+3.7** 3/3
Pixel prediction	8.9	445 444–445	−1.8 1/3	−3.3 3/3	+0.6 3/3	+1.5 2/3	−2.4 2/3	+1.3 2/3	−7.7 2/3	−7.1 3/3	+1.6 3/3	−1.6 1/3	−2.7 3/3	+1.0 2/3
Contrastive prediction	96.0 95–97	293 286–304	**−4.1** 3/3	−2.6 3/3	+0.7 3/3	−7.5 3/3	−1.4 2/3	+0.9 2/3	−21.7 3/3	−13.7 3/3	+3.2 3/3	−5.5 3/3	−3.6 3/3	+1.4 3/3

the next observation: its latent keeps little else (reconstruction error 0.95, against 0.98 for a latent holding only the signal), but even at $d = 100$ its signal and next-latent errors are two orders of magnitude below those of reconstruction. The combined objective avoids both failures: at $d = 100$ it reconstructs nearly as well as reconstruction alone (0.032 against 0.022), its next-latent error (0.022) is close to that of the contrastive objective (0.006) and far below that of reconstruction (1.03), and it recovers the signal at every size, even at $d = 10$ (7.5×10^{-4}, against 0.89).

5 Experiments with robotic manipulation data

5.1 Temporal structure benefits video world models

We first compare temporal objectives at equal reconstruction quality of the autoencoder, and find that they improve the world model but cost far more autoencoder training.

Data and objectives. We consider Bridge V2 (Walke et al., 2023), a dataset of about 60,000 videos of manipulation with a WidowX 250 robot arm, together with the actions of the arm; we train on 53,192 and hold out 6,872. Starting from the pretrained autoencoder of FLUX.1 (Black Forest Labs, 2024), which encodes each frame independently, we fine-tune the encoder and the decoder on pairs of nearby frames with four objectives: reconstruction alone, and its combination (Equation 1) with latent, pixel, or contrastive future prediction (Section 3), the latter with negatives drawn from other videos. Details of the objectives are provided in Appendix B.

Experimental design. A temporal objective changes both the structure of the latent and how well it reconstructs, so we fine-tune each autoencoder until its reconstruction loss on held-out frame pairs reaches the same value, and then freeze it, so that all latents support reconstructions of the same quality. On each frozen latent space we train the same action-conditioned diffusion world model (Section 3), and roll it out under recorded actions, on held-out Bridge V2 videos (in distribution) and on 479 trajectories of the AutoEval benchmark (Zhou et al., 2025b), recorded separately from Bridge V2 on setups that reproduce four of its tasks (out of distribution). We compare the generated frames with the recorded ones using pixel mean squared error (MSE), LPIPS (Zhang et al., 2018) and SSIM (Wang et al., 2004). In distribution, we also evaluate the world model with few sampling steps, and over long horizons. Details on the protocol, architectures, training and evaluation are provided in Appendices C to F.

Temporal structure improves the world model, but at higher cost. At equal reconstruction, all three temporal objectives improve the world model. Latent prediction gives the largest gains, on 11 of

the 12 metrics, followed by contrastive prediction, while pixel prediction stays closer to the baseline. Latent prediction lowers the pixel MSE most in the harder settings: by 21.9% with two sampling steps, 13.8% out of distribution and 9.1% at the end of long rollouts, against 3.0% in distribution. Latent prediction also improves FID and FVD (Table 4), and its world model stays ahead of that of reconstruction alone throughout training (Figure 5). These gains come at a cost: to reach the reconstruction target, the autoencoder needs 33 to 38.5 times as many updates with latent prediction as with reconstruction alone, and about 22 times as many with contrastive prediction.

5.2 Improving the information-structure frontier

The cost of Section 5.1 points to a trade-off between the information a latent retains and its temporal structure. We characterize it, and show that conditioning the decoder on a past frame improves it.

Past-conditioned decoder. Similarly to the synthetic setting (Section 4), latent and contrastive prediction concentrate the latent variance: their latents need fewer principal components to explain 95% of their variance than those of reconstruction alone, less than half as many for latent prediction (Table 1). This competes with reconstruction, with a balance set by λ. To ease the tension, we propose to condition the decoder on the previous frame, $\hat{x}_{t+1} = D(z_{t+1}, x_t)$, which relieves the latent of carrying every detail of the frame. Similar ideas have appeared in prior work (Wu et al., 2023; Micheli et al., 2024; Wu et al., 2024; Tian et al., 2024; Fan et al., 2025; FSVideo Team, 2026; Li et al., 2021), where it serves compression, reconstruction quality, or the dynamics model's focus on what changes; here we use it to improve the information-structure frontier. We implement it by adding cross-attention layers to the pretrained decoder, attending to the previous frame and initialized so that the decoder starts as the pretrained one. In world-model rollouts, the decoder is conditioned on the initial context frame (Figure 1; Appendix C).

Experimental design. We fine-tune the pretrained autoencoder on pairs of consecutive frames with latent future prediction, the objective with the largest gains in Section 5.1, at weights $\lambda \in \{0, 0.1, 0.3, 0.5, 0.7\}$, with the ordinary and with the past-conditioned decoder, for 80,000 updates each. We measure the information retained by the latent with the reconstruction error on held-out frame pairs, and its structure with the unpredictable share of the latent variance: the error of an action-conditioned linear predictor fitted on frozen latents, evaluated on held-out clips and divided by the total variance of the target latents (Appendix E). At each weight, we also train the world model of Section 5.3 on a past-conditioned autoencoder and compare it with the one at $\lambda = 0$ (Appendix H.9).

A past-conditioned decoder improves the frontier. With the ordinary decoder, raising λ trades reconstruction for structure: from $\lambda = 0$ to 0.7, the unpredictable share falls from 0.21 to 0.07, while the reconstruction error rises to 54% above that of reconstruction alone. The past-conditioned decoder shifts the whole frontier, lowering the reconstruction error after 80,000 updates by 17 to 26% at every weight, and the error of its own latent predictor as well, at every checkpoint of training (Figure 8). Up to $\lambda = 0.5$, it thus reconstructs better than the ordinary decoder trained for reconstruction alone. Interventions confirm that it uses the past frame (Appendix H.4). On the world model, every weight improves few-step sampling and out-of-distribution prediction over $\lambda = 0$, and weights from 0.5 also improve long rollouts.

5.3 Temporal structure improves world models at equal training budget

In this section, we show that, at a fixed training budget, the past-conditioned decoder combined with a temporal objective gives better world models than reconstruction alone.

Experimental design. We run a second experiment that fixes the training budget instead of the reconstruction target: every autoencoder is fine-tuned for 80,000 updates on pairs of consecutive frames, and the same world model, smaller than that of Section 5.1, is trained for 100,000 updates on its latents (Appendix G). We consider latent prediction, the objective with the largest gains in Section 5.1, and inverse dynamics (Section 3), which uses the actions as targets rather than inputs and does not train the latent to be predictable. Both use the past-conditioned decoder and $\lambda = 0.7$, the largest weight of Section 5.2, where the objective shapes the latent most: at this weight the past-conditioned autoencoder still reconstructs worse than the baseline (Figure 3), so any gain of the

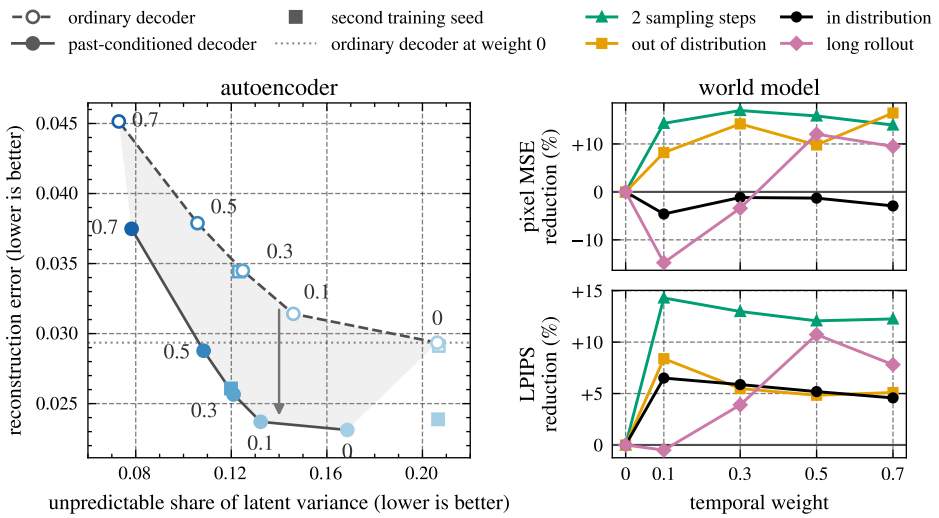


Figure 3: **The information-structure frontier and the world model. Left**: reconstruction error against the unpredictable share of the latent variance of autoencoders fine-tuned for 80,000 updates with latent prediction at weight λ, with the ordinary decoder or the past-conditioned decoder. **Right**: relative reduction of pixel MSE and LPIPS of the world model on each past-conditioned autoencoder against $\lambda = 0$. SSIM: Figure 24; autoencoder errors over training: Figure 8. One training seed per weight, and a second at $\lambda = 0$ and 0.3, which agrees.
Raising λ trades reconstruction for structure. The past-conditioned decoder improves the resulting Pareto frontier. Every weight improves the world model with few sampling steps and out of distribution, and from $\lambda = 0.5$ also on long rollouts.

world model comes despite a reconstruction cost (Appendix H.9 varies the weight). The baseline is the autoencoder fine-tuned with reconstruction alone and the ordinary decoder. We compare with reconstruction alone with the past-conditioned decoder, which separates the effect of the decoder from that of the temporal objective, and with the pretrained autoencoder without fine-tuning. In addition to the metrics of Section 5.1, we measure how predictable each latent space is for its world model, as the error of its one-step forecast of the next latent.

The past-conditioned decoder and temporal objectives improve the world model. The past-conditioned decoder alone improves every error metric of Table 2, and the temporal objectives add substantially to this gain, on almost every metric. Out of distribution, the decoder alone lowers the pixel MSE by 13.0%, and together with latent prediction or inverse dynamics by 23.6% and 22.7%; at the end of long rollouts, the objective contributes most of the gain (example rollouts in Figure 10). Both make the latent easier to forecast for the world model: the one-step forecast error falls by 80.3% with latent prediction and by 32.9% with inverse dynamics, although it is not trained for this, against 8.1% with the decoder alone. Their latents are also lower dimensional, spatially smoother and straighter over time than those of reconstruction alone (Appendix H.8). The world model also trains faster on these latents, reaching the final validation LPIPS of the baseline 7.6 and 8.8 times sooner, against 1.8 times with the decoder alone, and with 2 sampling steps both beat the LPIPS and SSIM of the baseline with 10 (Figure 6; SSIM in Figure 7). Among the distribution metrics, inverse dynamics lowers FID by 49.7% and FVD by 15.2% (Table 5); latent prediction raises FVD at $\lambda = 0.7$, but lowers both FID and FVD at every weight from 0.1 to 0.5 (Appendix H.9). With the past-conditioned world models trained twice as long, both objectives keep their gains out of distribution and in policy evaluation (Appendix H.7). In ablations (Appendix H.9), a KL penalty on the autoencoder leaves the picture unchanged; with a larger world model, latent prediction keeps its gains out of distribution and with few sampling steps; and temporal structure also complements a semantic encoder: with DINOv2 (Oquab et al., 2024) in place of the FLUX encoder, latent prediction at $\lambda = 0.1$ improves every metric.

Table 2: **Temporal objectives at equal training budget.** Fine-tuned autoencoders: 80,000 updates; world models: 100,000 updates on their latents. First row: the baseline (reconstruction alone, ordinary decoder), mean and standard error over 3 seeds (MSE $\times 10^3$). Other rows: relative change (%) against the baseline, mean over seeds, and number of seeds that improve; green: improved, orange: worsened; bold: best in each column. Latent forecast: error of the world model's one-step forecast of the next latent. Real-world agreement (absolute values): Pearson correlation between policy success rates in the world model and in the real world (WorldGym), with 10 and 2 sampling steps; one training seed; bootstrap standard error over tasks.
The past-conditioned decoder improves every error metric, and the temporal objectives add to it on almost every one.

	Latent forecast		In distribution			Out of distribution			2 sampling steps			Long rollout, steps 51–59			Real-world agreement	
	In	Out	MSE	LPIPS	SSIM	MSE	LPIPS	SSIM	MSE	LPIPS	SSIM	MSE	LPIPS	SSIM	10	2
Baseline (reconstruction)	0.232 ±0.008	0.186 ±0.005	14.34 ±0.16	0.194 ±0.002	0.779 ±0.001	36.15 ±0.98	0.302 ±0.003	0.682 ±0.004	18.35 ±0.65	0.232 ±0.004	0.756 ±0.003	47.93 ±1.51	0.394 ±0.002	0.595 ±0.004	0.51 ±0.10	0.21 ±0.07
Pretrained (no fine-tuning)	+9.6 0/3	+9.5 0/3	+3.3 0/3	+7.0 0/3	−2.7 0/3	−3.3 2/3	+2.3 2/3	−1.3 2/3	+4.4 1/3	+7.8 0/3	−2.9 0/3	−2.1 1/3	−0.2 1/3	−0.9 1/3	0.57 ±0.07	0.21 ±0.09
Past-conditioned decoder																
Reconstruction	−8.1 2/3	−6.9 2/3	−2.3 2/3	−9.2 3/3	+3.0 3/3	−13.0 3/3	−10.6 3/3	+5.4 3/3	−7.4 3/3	−14.6 3/3	+4.3 3/3	−6.8 3/3	−7.1 3/3	+5.1 3/3	0.49 ±0.08	0.15 ±0.07
Latent prediction	**−80.3** 3/3	**−73.7** 3/3	−0.4 2/3	−13.0 3/3	+4.0 3/3	**−23.6** 3/3	−14.8 3/3	+8.5 3/3	−14.6 3/3	−23.7 3/3	+6.6 3/3	**−18.4** 3/3	**−16.0** 3/3	**+11.3** 3/3	**0.62** ±0.07	**0.67** ±0.07
Inverse dynamics	−32.9 3/3	−29.7 3/3	**−8.8** 3/3	**−17.8** 3/3	**+4.4** 3/3	−22.7 3/3	**−21.7** 3/3	**+9.3** 3/3	**−26.6** 3/3	**−30.2** 3/3	**+7.9** 3/3	−17.9 3/3	−15.4 3/3	+9.8 3/3	0.57 ±0.07	0.51 ±0.09

Policy evaluation. In WorldGym (Quevedo et al., 2026), robot policies act inside a world model and a vision-language model judges task success. We run OpenVLA (Kim et al., 2025), Octo (Ghosh et al., 2024) and RT-1-X (O'Neill et al., 2024) on its 17 Bridge tasks inside the five world models of Table 2, and measure the agreement with the real world as the Pearson correlation between simulated and real success rates over the 51 pairs of policy and task (Appendix H.5). With 10 sampling steps, both temporal world models agree with the real world at least as well as the reconstruction-only ones (0.62 and 0.57, against 0.49 to 0.57). With 2 steps, the agreement of every reconstruction-only world model collapses, to 0.15 to 0.21, while latent prediction keeps 0.67 and inverse dynamics 0.51. The drop comes from the comparison between policies: OpenVLA, by far the strongest policy in the real world (Kim et al., 2025), falls from about 20% to 3 to 6% success in the reconstruction-only world models, below Octo (Figure 9; per-policy correlations in Table 8).

6 Discussion

Our results show that temporal structure in the latent space of video world models yields large practical benefits for robotic manipulation: the same generation quality with five times fewer sampling steps, the same perceptual quality with eight times shorter training, better generalization out of distribution and extrapolation to longer horizon than seen during training, and policy evaluations that agree better with the real world. Temporal structure should therefore be treated as an important design criterion for the latent spaces of video world models. Additionally, our results indicate that the choice between generative and reconstruction-free world models need not be a dichotomy of irreconcilable alternatives: one can retain the ability to generate realistic rollouts to interact with powerful VLM judges and policies, while imposing useful temporal structure that makes latents easily predictable from the past. With complex real-world data, this was helped by a simple hierarchy through the past-conditioned decoder design; it would be interesting to consider learning deeper hierarchies of dynamical representations, to enable flexible inference that trades off temporal and state resolution for horizon at a fixed budget.

Limitations. Our experiments cover a synthetic environment and robot manipulation videos, with a pretrained image autoencoder and a semantic encoder. Extending them to larger and more diverse video data, to tokenizers that also compress time and, as discussed, to hierarchical representations, is a natural direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools for method implementation: writing and editing the research code, including the code that produces the figures and tables. We have not used generative AI tools for theoretical or conceptual development, hypotheses, methodology or experiment design, synthetic-data generation, dataset cleaning or reformatting, qualitative analysis, or interpretation of results, and mathematical claims or proofs and translation are not applicable to this work. Additionally, we used generative AI tools for literature search and citation checking, for drafting parts of the appendix, and for editing the manuscript for readability. A vision-language model, Claude Opus 5.5, is also a component of one evaluation: it judges task success in the policy evaluation of Section 5.3 (Appendix H.5). We have reviewed all AI-assisted work: AI-assisted code was verified through tests and experiment-level checks, citations were checked against the cited works, and every edit to the manuscript was reviewed by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 and Appendices B to E specify the objectives, architectures, training and evaluation of the video experiments, Appendices F and G give the settings of the two main comparisons, and Appendix A specifies the synthetic study, including its data. The supplementary material contains anonymized source code for fine-tuning the autoencoders, training the world models and evaluating them, together with rollout videos of the world models.

REFERENCES

Pulkit Agrawal, Ashvin V. Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to Poke by Poking: Experiential Learning of Intuitive Physics. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for World Modeling: Visual Details Matter in Atari. In *Advances in Neural Information Processing Systems*, volume 37, pp. 58757–58791, 2024.

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. URL https://arxiv.org/abs/2506.09985.

Randall Balestriero and Yann LeCun. How Learning by Reconstruction Produces Uninformative Features For Perception. In *International Conference on Machine Learning*, pp. 2566–2585, 2024.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and Scalable Self-Supervised Learning Without the Heuristics, November 2025. URL https://arxiv.org/abs/2511.08544.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. In *International Conference on Learning Representations*, 2022. doi: 10.48550/arXiv.2105.04906.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting Feature Prediction for Learning Visual Representations from Video. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=QaCCuDfBk2.

Jai Bardhan, Patrik Drozdík, Josef Sivic, and Vladimir Petrik. Persistent Robot World Models: Stabilizing Multi-Step Rollouts via Reinforcement Learning. In *European Conference on Computer Vision*, 2026.

Black Forest Labs. FLUX. GitHub repository, 2024. URL https://github.com/black-forest-labs/flux. Accessed 2026-09-17.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22563–22575, 2023. doi: 10.48550/arXiv.2304.08818.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative Interactive Environments. In *International Conference on Machine Learning*, 2024. doi: 10.48550/arXiv.2402.15391.

João Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token Prediction Meets Full-Sequence Diffusion. In *Advances in Neural Information Processing Systems*, volume 37, pp. 24081–24125, 2024. URL https://openreview.net/forum?id=yDo1ynArjj.

Sixiang Chen, Jiaming Liu, Jixian Wu, Yichen Guo, Tinghao Wang, Siyuan Qian, Hao Chen, Jiajun Cao, Jian Tang, and Shanghang Zhang. Do Robotic World Models Really Follow Actions? Diagnosing and Aligning Action-Conditioned Generation for Policy Learning, 2026. URL https://arxiv.org/abs/2608.24885.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision Transformers Need Registers. In *International Conference on Learning Representations*, 2024.

Decart, Julian Quevedo, Quinn McIntyre, Spruce Campbell, Xinlei Chen, and Robert Wachen. Oasis: A Universe in a Transformer, October 2024. URL https://oasis-model.github.io/. Accessed 2026-04-11.

Fei Deng, Ingook Jang, and Sungjin Ahn. DreamerPro: Reconstruction-Free Model-Based Reinforcement Learning with Prototypical Representations. In *International Conference on Machine Learning*, pp. 4956–4975, 2022.

Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Kaelbling, Andy Zeng, and Jonathan Tompson. Video Language Planning. In *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2310.10625.

Peter Dürr, Mireille El Gheche, Guilherme Jorge Maeda, Nobuhiko Mukai, Naoya Takahashi, Stefan Heusser, Hamdi Sahloul, Yamen Saraiji, Pavel Adodin, Yin Bi, Sam Blakeman, Christian Conti, Dunai Fuentes Hitos, Yunpu Hu, Farshad Khadivar, Raphaela Kreiser, Luz Martinez, Fabian Schilling, Ricardo Tapiador Morales, Guillem Torrente, Mario Ynocente Castro, Lison Abecassis, Alberto Giammarino, Yu-Ting Huang, Yannik Nagel, Andrea Scotti, Alexander Sigrist, Tiago Silva, Etienne Walther, Jengyan Wong, Bilan Yang, Asude Aydin, Divij Grover, Apurv Saha, Valentina Cavinato, Takekazu Kakinuma, Taishi Kunori, Valentin Monferrato, Stefan Richter, Stefanos Charalambous, Simon Guist, Mads Alber Kuhlmann-Jorgensen, Lorenzo Miele, Agis Politis, Mattia Scardecchia, Hiroaki Kitano, Peter R. Wurman, Peter Stone, and Michael Spranger. Outplaying elite table tennis players with an autonomous robot. *Nature*, 652(8111):886–891, April 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10338-5.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming Transformers for High-Resolution Image Synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021.

Xiang Fan, Xiaohang Sun, Kushan Thakkar, Zhu Liu, Vimal Bhat, Ranjay Krishna, and Xiang Hao. RefTok: Reference-Based Tokenization for Video Generation, 2025. URL https://arxiv.org/abs/2507.02862.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised Learning for Physical Interaction through Video Prediction. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

FSVideo Team. FSVideo: Fast Speed Video Diffusion Model in a Highly-Compressed Latent Space, 2026. URL https://arxiv.org/abs/2602.02092.

Gemini Robotics Team. Evaluating Gemini Robotics Policies in a Veo World Simulator, 2025. URL https://arxiv.org/abs/2512.10675.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. In *Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.090.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. doi: 10.48550/arXiv.2006.07733.

Minghui Guo, Shengqiong Wu, and Hao Fei. V-RAE: Rethinking Video Latent Spaces for Generation, August 2026a. URL https://arxiv.org/abs/2608.13556.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A Controllable Generative World Model for Robot Manipulation. In *International Conference on Learning Representations*, 2026b.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. LTX-Video: Realtime Video Latent Diffusion, 2024. URL https://arxiv.org/abs/2501.00103.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning Latent Dynamics for Planning from Pixels, 2018. URL https://arxiv.org/abs/1811.04551.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering Diverse Control Tasks through World Models. *Nature*, 640(8059):647–653, 2025a. doi: 10.1038/s41586-025-08744-2.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training Agents Inside of Scalable World Models, September 2025b. URL https://arxiv.org/abs/2509.24527.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, Robust World Models for Continuous Control. In *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2310.16828.

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. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A Generative World Model for Autonomous Driving, September 2023. URL https://arxiv.org/abs/2309.17080.

Anthony Hu, Václav Volhejn, Adrien Ramanana Rahary, Chris Mulder, Aditya Makkar, Alyx Liao, Amélie Royer, Manu Orsini, Adam Jelley, Eloi Alonso, Florian Laurent, Fredrik Norén, James Swingos, Jan Hünermann, Kent Rollins, Lucas Hosseini, Matthieu Le Cauchois, Maxim Peter, Pim de Witte, Tim Brown, Vincent Micheli, Moritz Böhle, Gabriel de Marmiesse, Viktoriia Sharmanska, Lucia Specia, Michael Black, and Patrick Pérez. Multiplayer Interactive World Models with Representation Autoencoders, July 2026. URL https://arxiv.org/abs/2607.05352.

Junchao Huang, Ziyang Ye, Xinting Hu, Tianyu He, Guiyu Zhang, Shaoshuai Shi, Jiang Bian, and Li Jiang. LIVE: Long-horizon Interactive Video World Modeling. In *International Conference on Machine Learning*, 2026a.

Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2World: Crafting Video Diffusion Models to Interactive World Models. In *International Conference on Learning Representations*, 2026b. doi: 10.48550/arXiv.2505.14357.

Thomas Hubert, Rishi Mehta, Laurent Sartran, Miklós Z. Horváth, Goran Žužić, Eric Wieser, Aja Huang, Julian Schrittwieser, Yannick Schroecker, Hussain Masoom, Ottavia Bertolli, Tom Zahavy, Amol Mandhane, Jessica Yung, Iuliya Beloshapka, Borja Ibarz, Vivek Veeriah, Lei Yu, Oliver Nash, Paul Lezeau, Salvatore Mercuri, Calle Sönne, Bhavik Mehta, Alex Davies, Daniel Zheng, Fabian Pedregosa, Yin Li, Ingrid von Glehn, Mark Rowland, Samuel Albanie, Ameya Velingker, Simon Schmitt, Edward Lockhart, Edward Hughes, Henryk Michalewski, Nicolas Sonnerat, Demis Hassabis, Pushmeet Kohli, and David Silver. Olympiad-level formal mathematical reasoning with reinforcement learning. *Nature*, 651(8106):607–613, March 2026. ISSN 1476-4687. doi: 10.1038/s41586-025-09833-y.

Allan Jabri, David J. Fleet, and Ting Chen. Scalable Adaptive Computation for Iterative Generation. In *International Conference on Machine Learning*, pp. 14569–14589, 2023.

Byeongguk Jeon, Seonghyeon Ye, JaeHyeok Doo, Sungdong Kim, Minjoon Seo, Hyungmok Son, and Kimin Lee. RoboWorld: Fast and Reliable Neural Simulators for Generalist Robot Policy Evaluation, 2026. URL https://arxiv.org/abs/2607.01060.

Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How Far is Video Generation from World Model: A Physical Law Perspective. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2411.02385.

Markus Karmann, Shile Li, Christian Internò, Bruno Andreis, David Klindt, Randall Balestriero, Jindong Gu, Philip Torr, Qi Zhang, Peng-Tao Jiang, Hao Zhang, Bo Li, and Onay Urfalioglu. MotionJEPA: Preventing Temporal Feature Collapse by Capturing Visual Changes in Latent Space, September 2026. URL https://arxiv.org/abs/2609.23881.

Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Champion-level drone racing using deep reinforcement learning. *Nature*, 620 (7976):982–987, August 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06419-4.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model. In *Conference on Robot Learning*, pp. 2679–2713, 2025.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6114.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. HunyuanVideo: A Systematic Framework For Large Video Generative Models, 2024. URL https://arxiv.org/abs/2412.03603.

Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. EQ-VAE: Equivariance Regularized Latent Space for Improved Generative Image Modeling. In *International Conference on Machine Learning*, 2025. doi: 10.48550/arXiv.2502.09509.

Yann LeCun. A Path Towards Autonomous Machine Intelligence. OpenReview, 2022. URL https://openreview.net/forum?id=BZ5a1r-kVsf. Version 0.9.2, 2022-06-27.

Jiahao Li, Bin Li, and Yan Lu. Deep Contextual Video Compression, 2021. URL https://arxiv.org/abs/2109.15047.

Zongjian Li, Bin Lin, Yang Ye, Liuhan Chen, Xinhua Cheng, Shenghai Yuan, and Li Yuan. WF-VAE: Enhancing Video VAE by Wavelet-Driven Energy Flow for Latent Video Diffusion Model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17778–17788, 2025.

Etai Littwin, Omid Saremi, Madhu Advani, Vimal Thilak, Preetum Nakkiran, Chen Huang, and Joshua Susskind. How JEPA Avoids Noisy Features: The Implicit Bias of Deep Linear Self Distillation Networks. In *Advances in Neural Information Processing Systems*, volume 37, pp. 91300–91336, 2024.

Shizhan Liu, Xinran Deng, Zhuoyi Yang, Jiayan Teng, Xiaotao Gu, and Jie Tang. Delving into Latent Spectral Biasing of Video VAEs for Superior Diffusability. In *European Conference on Computer Vision*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019.

Susie Lu, Haonan Chen, Weirui Ye, and Yilun Du. DriftWorld: Fast World Modeling through Drifting, 2026. URL https://arxiv.org/abs/2607.15065.

Jiayi Luo, Hanxin Zhu, Chen Gao, Jiankun Wang, Cong Wang, Tianyu He, Jianxin Li, and Zhibo Chen. EmbodiedVAE: Disentangled Video VAE for Efficient and Controllable Embodied Manipulation. In *European Conference on Computer Vision*, 2026.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable End-to-End Joint-Embedding Predictive Architecture from Pixels, March 2026. URL https://arxiv.org/abs/2603.19312.

Vincent Micheli, Eloi Alonso, and François Fleuret. Efficient World Models with Context-Aware Tokenization, 2024. URL https://arxiv.org/abs/2406.19320. ICML 2024.

Ofir Nachum and Mengjiao Yang. Provable Representation Learning for Imitation with Contrastive Fourier Features. In *Advances in Neural Information Processing Systems*, pp. 30100–30112, 2021. doi: 10.48550/arXiv.2105.12272.

Zaruhi Navasardyan, Tatul Danielyan, and Hrant Davtyan. FailBench: How Reliable are VLMs at Judging Robot Task Success?, September 2026. URL https://arxiv.org/abs/2609.03611.

Tung Nguyen, Rui Shu, Tuan Pham, Hung Bui, and Stefano Ermon. Temporal Predictive Coding For Model-Based Planning In Latent Space. In *International Conference on Machine Learning*, pp. 8130–8139, 2021.

Nilaksh, Saurav Jha, Artem Zholus, and Sarath Chandar. Reconstruction or Semantics? What Makes a Latent Space Useful for Robotic World Models, May 2026. URL https://arxiv.org/abs/2605.06388.

NVIDIA. Cosmos World Foundation Model Platform for Physical AI, 2025. URL https://arxiv.org/abs/2501.03575.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L. Lewis, and Satinder Singh. Action-Conditional Video Prediction using Deep Networks in Atari Games. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Masashi Okada and Tadahiro Taniguchi. Dreaming: Model-based Reinforcement Learning by Latent Imagination without Reconstruction. In *IEEE International Conference on Robotics and Automation*, pp. 4209–4215, 2021. doi: 10.1109/ICRA48506.2021.9560734.

Abby O'Neill et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In *IEEE International Conference on Robotics and Automation*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

OpenAI. Competitive Programming with Large Reasoning Models, 2025. URL https://arxiv.org/abs/2502.06807.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven Exploration by Self-supervised Prediction. In *International Conference on Machine Learning*, pp. 2778–2787, 2017.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *IEEE/CVF International Conference on Computer Vision*, pp. 4172–4182, 2023. doi: 10.48550/arXiv.2212.09748.

Ayan Pendharkar. Predictive Objectives Discard Exogenous Control-Relevant Features: A Controlled Mechanistic Study, June 2026. URL https://arxiv.org/abs/2606.30068.

Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. WorldGym: World Model as An Environment for Policy Evaluation. In *International Conference on Learning Representations*, 2026. doi: 10.48550/arXiv.2506.00613.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10674–10685, 2022. doi: 10.48550/arXiv.2112.10752.

Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. GAIA-2: A Controllable Multi-View Generative World Model for Autonomous Driving, March 2025. URL https://arxiv.org/abs/2503.20523.

Tim Salimans and Jonathan Ho. Progressive Distillation for Fast Sampling of Diffusion Models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved Techniques for Training GANs. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-Contrastive Networks: Self-Supervised Learning from Video. In *IEEE International Conference on Robotics and Automation*, pp. 1134–1141, 2018. doi: 10.1109/ICRA.2018.8462891.

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, February 2026. URL https://arxiv.org/abs/2602.02454.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A General Reinforcement Learning Algorithm That Masters Chess, Shogi, and Go through Self-Play. *Science*, 362(6419):1140–1144, 2018. doi: 10.1126/science.aar6404.

Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the Diffusability of Autoencoders. In *International Conference on Machine Learning*, 2025. doi: 10.48550/arXiv.2502.14831.

Vlad Sobal, Jyothir S, Siddhartha Jalagam, Nicolas Carion, Kyunghyun Cho, and Yann LeCun. Joint Embedding Predictive Architectures Focus on Slow Features, November 2022. URL https://arxiv.org/abs/2211.10831.

Vlad Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from Reward-Free Offline Data: A Case for Planning with Latent Dynamics Models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 43905–43941, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-Guided Video Diffusion. In *International Conference on Machine Learning*, pp. 56242–56280, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

Rui Tian, Qi Dai, Jianmin Bao, Kai Qiu, Yifan Yang, Chong Luo, Zuxuan Wu, and Yu-Gang Jiang. REDUCIO! Generating 1K Video within 16 Seconds using Extremely Compressed Motion Latents, 2024. URL https://arxiv.org/abs/2411.13552. ICCV 2025.

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, December 2025. URL https://arxiv.org/abs/2511.11520.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards Accurate Generative Models of Video: A New Metric & Challenges, 2018. URL https://arxiv.org/abs/1812.01717.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion Models Are Real-Time Game Engines. In *International Conference on Learning Representations*, 2025.

Hugues Van Assel, Mark Ibrahim, Tommaso Biancalani, Aviv Regev, and Randall Balestriero. Joint-Embedding vs Reconstruction: Provable Benefits of Latent Space Prediction for Self-Supervised Learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 21897–21937, 2025.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding, July 2018. URL https://arxiv.org/abs/1807.03748.

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 *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1723–1736, 2023. URL https://proceedings.mlr.press/v229/walke23a.html.

Wan Team. Wan: Open and Advanced Large-Scale Video Generative Models, 2025. URL https://arxiv.org/abs/2503.20314.

Junke Wang, Qihang Zhang, Shuai Yang, Yiming Luo, Yujun Shen, Zuxuan Wu, Yu-Gang Jiang, and Yinghao Xu. RepWAM: World Action Modeling with Representation Visual-Action Tokenizers, June 2026. URL https://arxiv.org/abs/2606.13674.

Tongzhou Wang and Phillip Isola. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939, 2020.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

Jialong Wu, Haoyu Ma, Chaoyi Deng, and Mingsheng Long. Pre-training Contextualized World Models with In-the-wild Videos for Reinforcement Learning, 2023. URL https://arxiv.org/abs/2305.18499. NeurIPS 2023.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. iVideoGPT: Interactive VideoGPTs are Scalable World Models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 68082–68119, 2024.

Zaishuo Xia, Yukuan Lu, Xinyi Li, Yifan Xu, and Yubei Chen. Cloning Deterministic Worlds: The Critical Role of Latent Geometry in Long-Horizon World Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32602–32612, 2026.

Zhihao Xie, Junfeng Wu, Xinting Hu, Junchao Huang, and Li Jiang. VideoRAE: Taming Video Foundation Models for Generative Modeling via Representation Autoencoders, July 2026. URL https://arxiv.org/abs/2607.14088.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning Interactive Real-World Simulators. In *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2310.06114.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer. In *International Conference on Learning Representations*, 2025.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. Generation: Taming Optimization Dilemma in Latent Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15703–15712, 2025. doi: 10.48550/arXiv.2501.01423.

Jingfeng Yao, Yuda Song, Yucong Zhou, and Xinggang Wang. Towards Scalable Pre-training of Visual Tokenizers for Generation. In *European Conference on Computer Vision*, 2026.

Jewon Yeom, Hanseul Kim, Jeongjae Park, Sungmok Jung, Jaejin Lee, and Taesup Kim. What Makes Video World Model Latents Action-Relevant: Prediction over Reconstruction, 2026. URL https://arxiv.org/abs/2606.07687.

Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language Model Beats Diffusion – Tokenizer is Key to Visual Generation. In *International Conference on Learning Representations*, 2024.

Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal M. Patel, Paul Pu Liang, Daniel Khashabi, Cheng Peng, Rama Chellappa, Tianmin Shu, Alan Yuille, Yilun Du, and Jieneng Chen. World-in-World: World Models in a Closed-Loop World. In *International Conference on Learning Representations*, 2026a. doi: 10.48550/arXiv.2510.18135.

Mingkun Zhang, Wangtian Shen, Fan Zhang, Haijian Qin, Zihao Pei, and Ziyang Meng. RAE-NWM: Navigation World Model in Dense Visual Representation Space. In *European Conference on Computer Vision*, 2026b.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Yian Zhao, Feng Wang, Qiushan Guo, Chang Liu, Xiangyang Ji, Jian Zhang, and Jie Chen. Video Generation with Predictive Latents, 2026. URL https://arxiv.org/abs/2605.02134.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion Transformers with Representation Autoencoders. In *International Conference on Learning Representations*, 2026. doi: 10.48550/arXiv.2510.11690.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World Models on Pre-trained Visual Features enable Zero-shot Planning. In *International Conference on Machine Learning*, pp. 79115–79135, 2025a.

Zhiyuan Zhou, Pranav Atreya, You Liang Tan, Karl Pertsch, and Sergey Levine. AutoEval: Autonomous Evaluation of Generalist Robot Manipulation Policies in the Real World. In *Conference on Robot Learning*, 2025b. URL https://arxiv.org/abs/2503.24278.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A Fine-Grained World Model for Robot Manipulation. In *IEEE/CVF International Conference on Computer Vision*, pp. 9834–9844, 2025.

GUIDE TO THE APPENDIX

Appendix A gives the synthetic study and its full results. For the video experiments, Appendices B–E specify the objectives, architectures, training and evaluation; Appendices F and G give the settings specific to the two main comparisons. Appendix H presents the controls, complete quantitative results and learning curves, followed by decoder interventions, policy evaluation, latent-space analysis and ablations. Each comparison keeps its own reference: reconstruction alone with the ordinary decoder measures the combined benefit of the architecture and objective, whereas reconstruction alone with the past-conditioned decoder isolates the added benefit of the objective.

A SYNTHETIC STUDY: SETTING AND PROTOCOL

This appendix gives the protocol of the synthetic study of Section 4.

Environment. A state is $x_t = (s_t, \xi_t)$, with a signal $s_t \in \mathbb{R}^2$ and distractors $\xi_t \in \mathbb{R}^{100}$, and the observation is the state itself. At the start of a trajectory, s_0 and ξ_0 are drawn from a standard normal distribution. At every step an action is drawn uniformly among the four vectors $\pm 0.3\, e_1$ and $\pm 0.3\, e_2$, and the signal moves by it, $s_{t+1} = s_t + a_t$. At every step, ξ_{t+1} is a new standard normal draw, independent of the past. A trajectory has two transitions. The training, validation and test sets hold 10,000, 1,000 and 1,000 trajectories, that is, 20,000 training transitions. Every distractor coordinate has unit variance, and the variance of a signal coordinate grows from 1 at the first state of a trajectory to 1.09 at the third, so that a constant prediction has a mean squared error of about 1.05 on the signal and about 1 on the observation.

Networks. Every network is a multilayer perceptron with one hidden layer of 256 units and ReLU activations, except the probe, whose hidden layer has 64 units. The encoder maps the observation to the mean and the log-variance of a Gaussian posterior over $z \in \mathbb{R}^d$; the training loss uses a sample from the posterior, and everything downstream uses the mean, as on real video (Section 3). The decoder of $\mathcal{L}_{\mathrm{rec}}$ maps a latent back to $\mathbb{R}^{102}$. The key network g of $\mathcal{L}_{\mathrm{temp}}$ maps the next observation and the one-hot action to $\mathbb{R}^d$. The dynamics model maps a latent and the one-hot action to a latent, the probe maps a latent to the signal, and the readout decoder maps a latent to the observation.

Objectives. The reconstruction term is the squared error averaged over the 102 coordinates, plus $\beta = 10^{-4}$ times the KL divergence from the posterior to the standard normal distribution, summed over the latent coordinates. The temporal term is the InfoNCE loss with the standardized current latent $q_t = \mathrm{LN}(z_t)$ as the query and standardized keys $y = \mathrm{LN}(g(x_{t+1}, a_t))$, scored by $-\|q_t - y\|^2/(\tau d)$, where LN is layer normalization without affine parameters and τ is a temperature. The standardization fixes the norms of the query and the keys, so that the score depends on their angle only and the temperature transfers across latent sizes. The negatives are the keys of the other 255 transitions of the batch, which are drawn from the whole training set and thus, with rare exceptions, from other trajectories. The negatives penalize latents that do not distinguish observations, so that a collapsed latent does not minimize the temporal term (Wang & Isola, 2020); any information it discards is discarded by the objective itself, not by a collapse. The temperature, $\tau = 1$, and the KL weight were selected on the validation next-signal error in preliminary runs and are fixed across the grid. At $\lambda = 1$ the loss is exactly the contrastive objective, and at $\lambda = 0$ exactly the objective of the variational autoencoder; the reconstruction and KL terms keep their relative weight at every λ.

Training. The four phases run in sequence, each on the frozen output of the previous ones: the encoder first, then the dynamics model, the probe and the readout decoder on the latents of the frozen encoder. The readout decoder is separate from the decoder of $\mathcal{L}_{\mathrm{rec}}$, so that reconstruction is measured in the same way at every λ, including $\lambda = 1$, where the decoder of $\mathcal{L}_{\mathrm{rec}}$ receives no gradient. Each phase runs 7,500 steps of AdamW at batch size 256 and learning rate 10^{-3}, without weight decay; the gradient norm is clipped at 1 in the encoder phase. The final weights are used, and the validation set serves for monitoring only. The dynamics model minimizes the squared error to the latent of the next observation, the probe the squared error to the signal, and the readout decoder the squared error to the observation; the encoder receives no gradient from any of them.

Evaluation. All errors are squared errors on the test transitions, averaged over the target coordinates and over transitions; lower is better. With ψ the probe, ρ the readout decoder and ϕ the dynamics model, the reconstruction error is between $\rho(z_t)$ and x_t, the current-signal error between $\psi(z_t)$ and s_t, the next-latent error between $\phi(z_t, a_t)$ and z_{t+1}, and the next-signal error between $\psi(\phi(z_t, a_t))$ and s_{t+1}. Each error is divided by the variance of its target on the test set of the same seed, so that a constant prediction of the mean gives 1. The target of the next-latent error is the latent itself, whose scale is learned and differs across sizes and objectives; its variance is estimated as half the mean squared distance between the latents of two independently drawn test observations. In the left panel of Figure 2, each error of an encoder is divided by the error of the full observation on the same seed, which shares its data; the panel shows the geometric mean of this ratio over seeds and the standard error of its logarithm.

Grid and replication. The grid crosses the latent sizes $d \in \{10, 20, 30, 50, 100\}$ with the weights $\lambda \in \{0, 0.05, 0.2, 0.5, 0.8, 1\}$; Section 4 shows $\lambda \in \{0, 0.5, 1\}$, and Figure 4 every weight. The full observation ($d = 102$) is the reference, which makes 31 configurations. Every configuration is repeated with three seeds, which set the data draw and every initialization, so that the replicates are independent draws of the whole experiment: 93 runs. Figure 4 reports the mean and the standard error over seeds. A run takes a few minutes on 8 CPU cores.

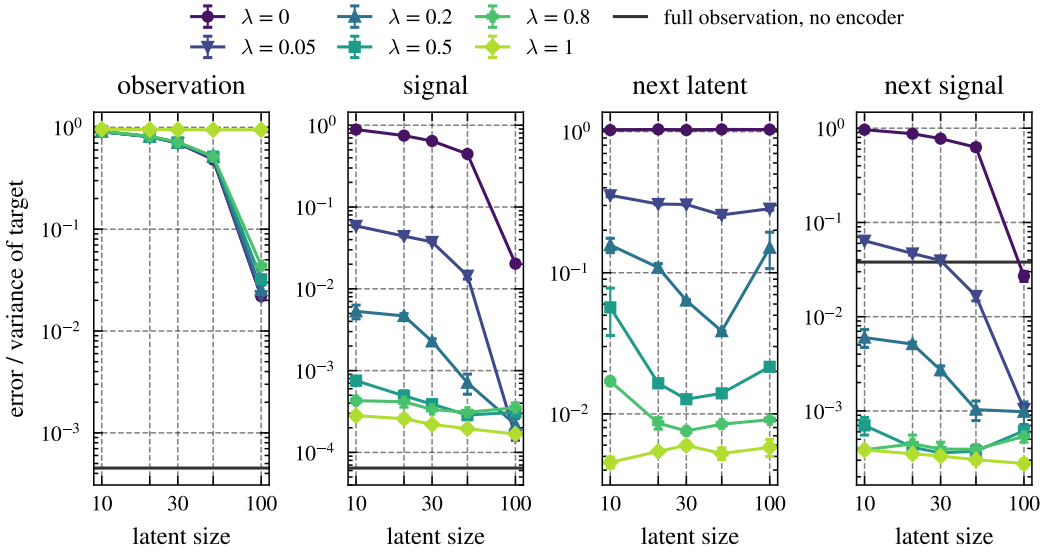


Figure 4: Full synthetic study: reconstruction, current-signal, next-latent and next-signal errors against latent dimension, for all six temporal weights λ. The temporal term is contrastive prediction; $\lambda = 0$ uses reconstruction alone and $\lambda = 1$ the temporal term alone. Horizontal lines give the full observation supplied to the same downstream networks without an encoder. Errors are divided by their target variance; lower is better. Mean and standard error over three independent training and data seeds. Intermediate weights retain reconstruction information while improving signal recovery and latent predictability over reconstruction alone.

Relation to the objectives on real video. The temporal term is the contrastive objective of Appendix B in its simplest form. The score is a squared distance between standardized vectors, without the bilinear form, the momentum encoder and the queue, and the roles are exchanged: the current latent is the query, and the keys are computed from the next observation and the action by a separate network (Nachum & Yang, 2021), instead of a prediction from the current latent being scored against encoded future frames. The reconstruction terms differ as the data do: a squared error with a KL penalty here, the ℓ_1 and LPIPS terms of Appendix B on real video.

B TEMPORAL OBJECTIVES AS IMPLEMENTED

This appendix gives the form of each objective of Section 3 as used in the experiments. The autoencoder is trained on pairs of frames x_t and x_{t+k}, with latents $z_t = E(x_t)$ and $z_{t+k} = E(x_{t+k})$, and the actions $a_{t:t+k-1}$ executed between them. The three forward objectives score the prediction $\hat{z}_{t+k} = P(z_t, a_{t:t+k-1})$ of the predictor of Appendix C. The reconstruction loss $\mathcal{L}_{\text{rec}}$ is the mean absolute pixel error plus VGG-based LPIPS (Zhang et al., 2018), with weight one on each term. The pixel term is averaged over channels and spatial positions on images in $[0, 1]$; LPIPS receives the images mapped to $[-1, 1]$. All losses below are averaged over examples. Unless stated otherwise, no adversarial or KL term is used, and the latent is the mean of the encoder's posterior throughout.

Latent future prediction. The prediction is regressed onto the future latent,

$$\mathcal{L}_{\text{latent}} = \frac{1}{d_z} \big\| P(z_t, a_{t:t+k-1}) - \text{sg}[z_{t+k}] \big\|_1, \tag{2}$$

where d_z is the number of latent coordinates (16,384 for FLUX.1) and sg is the stop-gradient operator; the target comes from the same encoder, without a momentum copy.

Pixel future prediction. The prediction is decoded and compared with the future frame,

$$\mathcal{L}_{\text{pixel}} = \mathcal{L}_{\text{rec}}\big(x_{t+k},\, D(P(z_t, a_{t:t+k-1}))\big), \tag{3}$$

with the same decoder and reconstruction loss as for the current frame; the gradient reaches the encoder, the predictor and the decoder.

Contrastive future prediction. The prediction is trained to identify the future frame among alternatives with the InfoNCE loss. Queries q and keys y are unit-normalized projections of the prediction and of frames encoded by a momentum copy of the encoder, compared through a learned bilinear form W:

$$\mathcal{L}_{\text{contr}} = -\log \frac{\exp(q^\top W y^+)}{\exp(q^\top W y^+) + \sum_{y^- \in \mathcal{N}} \exp(q^\top W y^-)}, \tag{4}$$

where y^+ is the key of x_{t+k} and $\mathcal{N}$ contains the keys of the future frames of the other videos in the batch and a queue of 40,960 keys from recent batches. The projection is a single residual 7×7 convolution; a momentum copy of it is applied to the keys together with the momentum encoder, whose decay is 0.9978. W is initialized at the identity, so that the score starts as a cosine similarity, and the temperature is one.

Inverse dynamics. A head H predicts the action between two consecutive frames from their latents,

$$\mathcal{L}_{\text{inv}} = \ell\big(H(z_t, z_{t+1}),\, a_t\big), \tag{5}$$

where ℓ averages, over the seven components, the squared error on the six translation and rotation components, standardized with the mean and standard deviation of the training actions, and the binary cross-entropy on the gripper state; both latents receive gradients.

Loss scaling. The implementation multiplies the convex combination of Equation 1 by two, $\mathcal{L} = 2\big[(1-\lambda)\,\mathcal{L}_{\text{rec}} + \lambda\,\mathcal{L}_{\text{temp}}\big]$, so that $\lambda = 1/2$ gives unit weight to both terms. The factor sets the scale of the gradients before gradient clipping.

Reconstruction target within a pair. With the ordinary decoder, the earlier frame of the pair is reconstructed from its own latent, $D(z_t) \approx x_t$, and the future latent z_{t+k} is computed without gradient. With the past-conditioned decoder, the later frame is reconstructed from its latent and the earlier frame, $D(z_{t+k}, x_t) \approx x_{t+k}$; the latent of the earlier frame then receives no reconstruction gradient, and the latent of the later frame receives no gradient from the forward objectives.

C ARCHITECTURES

We describe the networks shared by the FLUX-based video experiments below. The world-model size differs between the matched-reconstruction and matched-budget comparisons; the semantic-encoder ablation has additional changes specified in Appendix H.9.2.

Autoencoder. Unless stated otherwise, the real-video experiments initialize both the encoder and the decoder from the pretrained autoencoder of FLUX.1-schnell (Black Forest Labs, 2024). We use its variational autoencoder component, with 16 latent channels and a spatial downsampling factor of eight: a 256×256 frame gives a $16 \times 32 \times 32$ latent. Both the encoder and decoder are fine-tuned during representation learning and frozen during world-model training; the pretrained-autoencoder control skips fine-tuning.

Predictor. Two 1×1 convolutions map the 16 channels of the latent to a width of 512 and back, around two residual blocks. Each block applies a depthwise-separable convolution with kernel 3, self-attention over the 32×32 positions with rotary position embeddings (Su et al., 2024) extended to two dimensions and 8 heads, cross-attention to the action tokens, and a feed-forward layer with expansion factor 4 and GELU activation. The sublayers use group normalization with 32 groups, residual connections and no dropout. Each action is embedded by a two-layer perceptron of width 512 with GELU activation and a learned position embedding. Together with a learned summary token, these embeddings pass through a two-layer transformer of width 512, with 8 heads, feed-forward width 2,048, GELU activations and layer normalization before each sublayer and at the output. The action-token outputs serve as keys and values for cross-attention. The predictor has 15.6 M parameters, 6.6 M of them in the action encoder.

Inverse-dynamics head. The two latents are concatenated along channels and processed by the same blocks without cross-attention; the output is averaged over positions and mapped to the seven action components by a two-layer perceptron. The head has 7.1 M parameters.

Contrastive head. The projection applied to the prediction and, through its momentum copy, to the keys is a single residual 7×7 convolution that keeps the shape of the latent. The bilinear form W of Equation 4 acts on the flattened projections.

Past-conditioned decoder. The pretrained decoder (49.5 M parameters, all fine-tuned) receives five cross-attention adapters with 8 heads, after the middle block and after each of the four upsampling blocks (3.7 M parameters). The context is a 32×32 grid of 1,024 tokens from the previous frame. At each position, an 8×8 pixel patch and the corresponding middle-block features of a frozen copy of the pretrained encoder (34.3 M parameters, not trained) are separately projected to width 512, concatenated, and projected back to width 512 (0.9 M parameters). Each adapter projects these context tokens to the channel width of its decoder block for keys and values, and uses the decoder features as queries. The adapters use group normalization with 32 groups and two-dimensional rotary position embeddings on queries and keys, with no dropout. The output projection of every adapter is zero-initialized. The adapters and the projections add 4.6 M trained parameters; at inference, the decoder with its context encoder has 88 M parameters against 50 M for the ordinary decoder.

World-model transformer. Each standardized latent frame is patchified with patch size 2 into 256 tokens. Every block applies bidirectional self-attention within a frame, causal self-attention across frames at each spatial position, both with rotary position embeddings, and a feed-forward layer; adaptive layer normalization with zero initialization modulates the block from the sum of the noise-level embedding and a linear embedding of the action of that frame, so that the modulation is specific to each frame, as diffusion forcing requires; the action a_t reaches frame $t + 1$ through the causal attention across frames. The final layer predicts the velocity of every token. The matched-reconstruction experiment uses width 1,024, 16 layers and 16 heads; the matched-budget experiment uses width 384, 12 layers and 6 heads, 64.4 M parameters.

D TRAINING DETAILS

Data pipeline. Bridge V2 is recorded at 5 Hz (Walke et al., 2023). We use the TensorFlow Datasets release `bridge_dataset`, version 1.0.0, preserving its supplied training and validation splits: 53,192 training videos and 6,872 held-out videos. Frames are decoded from the stored videos, scaled to $[0, 1]$, and resized to 256×256 without cropping. The FLUX encoder maps these inputs to $[-1, 1]$; the DINOv2 preprocessing is given in Appendix H.9.2. The action a_t has seven components: the translation and the rotation of the end effector, and the gripper state. The translation and the rotation

components of the action are rescaled by fixed factors of 35 and 5.6 and clipped to ± 1.74 and ± 1.39; the gripper state is -1 when open and $+1$ when closed. For network conditioning, these seven values are followed by three zero entries and a null-action flag, giving 11 input components. The flag is zero for recorded actions; when conditioning is dropped, the flag is one and all other components are zero. For autoencoder training, one pair is drawn per video and epoch with a uniformly distributed start frame. The trade-off and matched-budget experiments use consecutive frames ($k = 1$); the matched-reconstruction experiment draws a triplet of consecutive frames and uses its first two, its last two or its outer two frames as the pair, with equal probability, so that $k = 1$ with probability $2/3$ and $k = 2$ with probability $1/3$. For world-model training, one clip of ten consecutive frames is drawn per video and epoch with a uniformly distributed start frame. Videos with fewer than ten frames are excluded; the matched-budget world models use 53,088 training videos.

Autoencoder validation. The matched-reconstruction experiment resamples one held-out pair per video at each validation pass, using the same pair-sampling rule as in training. The trade-off and matched-budget experiments instead use a fixed set of 6,872 consecutive held-out pairs, one per video, drawn with data seed 42 and shared across objectives. In the matched-reconstruction experiment, the held-out videos that determine the stopping point also provide the in-distribution evaluation clips; stopping uses only the autoencoder's reconstruction loss, with the same threshold for every objective.

Autoencoder optimization. All experiments use AdamW (Loshchilov & Hutter, 2019) with $\beta = (0.9, 0.999)$ and no weight decay, a linear warm-up of 1,000 updates, gradient clipping at 0.5, and bf16 mixed precision, on one H200 GPU. The matched-reconstruction experiment uses a batch of 24 pairs and a learning rate of 2.4×10^{-6}, and trains until the reconstruction loss on the held-out pairs reaches 0.0345, checked at the end of every epoch of 2,215 updates. The trade-off and matched-budget experiments use a batch of 48 pairs and a learning rate of 3.46×10^{-6}, train for 80,000 updates, and use the final checkpoint.

Diffusion process. Each latent is corrupted independently as $z_t^{k_t} = \sqrt{\bar{\alpha}_{k_t}}\, z_t + \sqrt{1 - \bar{\alpha}_{k_t}}\, \epsilon_t$, with $\epsilon_t \sim \mathcal{N}(0, I)$, k_t uniform on $\{1, \dots, K\}$, and a variance-preserving schedule $1 > \bar{\alpha}_1 > \cdots > \bar{\alpha}_K \approx 0$ (Ho et al., 2020). A network v_θ receives the noisy latents, their noise levels, and the actions, and regresses the velocity $v_t = \sqrt{\bar{\alpha}_{k_t}}\, \epsilon_t - \sqrt{1 - \bar{\alpha}_{k_t}}\, z_t$ (Salimans & Ho, 2022) of every frame:

$$\mathcal{L}_{\mathrm{DF}}(\theta) = \mathbb{E}\left[\sum_{t=1}^{T} \left\| v_\theta\big(z_{1:T}^{k_{1:T}}, k_{1:T}, a_{1:T}\big)_t - v_t \right\|_2^2\right]. \tag{6}$$

For the FLUX-based world models, the diffusion process has $K = 1{,}000$ noise levels with a sigmoid schedule (Jabri et al., 2023). The DINOv2 parameterization and noise-level sampling are specified in Appendix H.9.2. The sigmoid argument runs from -3 to 3 with temperature one, and the per-step noise variances β_k are clipped to $[0, 0.999]$. The squared velocity error is averaged over latent coordinates, frames and examples. The action is replaced by the null action with probability 0.1 during training, which allows classifier-free guidance (Ho & Salimans, 2021) at sampling time. The transformer has width 1,024 and 16 layers in the matched-reconstruction experiment and width 384 and 12 layers, 64.4 M parameters, in the matched-budget experiment (Appendix C).

World-model optimization. All experiments use AdamW with $\beta = (0.9, 0.99)$ and weight decay 0.02, bf16 mixed precision, and an exponential moving average of the weights with decay 0.999, which is used for sampling. The matched-reconstruction experiment uses a batch of 4 clips without gradient accumulation, a constant learning rate of 3×10^{-5}, and 238,896 updates, that is, 18 epochs. The matched-budget experiment uses an effective batch of 16 clips, a learning rate of 7×10^{-5} after a linear warm-up from 10^{-7} over 3,000 updates, gradient-norm clipping at 0.5, and 100,000 updates. Training uses 8 clips per step with two accumulation steps on one GPU; the two-GPU configuration of the larger world models is specified below.

Latent normalization. After fine-tuning, statistics of the encoder output are computed on 100 batches of training frames, one random frame per clip. The matched-reconstruction experiment divides the latent by one global standard deviation; the matched-budget experiment subtracts a per-channel mean and divides by a per-channel standard deviation. The statistics are stored with the world model, which is trained and sampled in the standardized space; the transformation is inverted before decoding.

Compute. The matched-reconstruction world models and the fine-tuned models of the matched-budget experiment use one H200 GPU per run. The width-1,024 models in the size ablation use two H200s with 8 clips per GPU and no accumulation. In the matched-budget experiment, fine-tuning an autoencoder for 80,000 updates takes about 42 hours, training a world model for 100,000 updates about 17 hours, and the complete evaluation of one world model about 15 hours.

E EVALUATION DETAILS

Evaluation sets. The in-distribution set consists of 2,000 clips of ten frames: one clip per video from a random permutation of the held-out videos, with a start frame drawn uniformly within the video, both with seed 42, so that every model is evaluated on the same clips. The sampling-step sweep uses the first 512 of these clips. The long-rollout evaluation uses 716 held-out videos, all 224 videos with at least 60 frames and 512 drawn uniformly among the videos with at least 20 frames, 20 of which are also among the 224, each rolled out once from its first frame for up to 59 predicted frames. The out-of-distribution set consists of all 479 AutoEval trajectories: 80 for closing the drawer, 94 for opening the drawer, 193 for placing the eggplant in the sink, and 112 for placing it in the basket. Frames are resized to 256×256; the rollout starts at frame index 20 (zero-based) of each trajectory and predicts the following 29 frames from that context frame. The translation and rotation components of the recorded actions are rescaled with the factors of Appendix D; the gripper command is binarized at 0.5 and mapped to the sign convention of Bridge V2, which is opposite to that of AutoEval.

Sampling. Table 3 gives the four evaluation conditions. DDIM uses the stated number of steps on a uniform grid over the 1,000 noise levels, without added noise, and classifier-free guidance with scale 3 at every step; the DINOv2 ablation uses the warped grid of Appendix H.9.2. For a chunk of eight future frames, all eight are appended as noise and denoised jointly at staggered noise levels, each frame initially at a higher level than the previous one; the final chunk can contain fewer frames. The past-conditioned decoder uses the initial real context frame throughout the rollout. Encoding and sampling run in bf16, decoding and the metrics in fp32. Within each comparison, the random state is reset before every model, so that all models use the same diffusion noise on the same clips; one rollout is drawn per clip and sampling seed.

Table 3: Evaluation conditions. Every model is evaluated on the same episodes with paired diffusion-noise seeds. Each rollout starts from one real context frame; scored prediction steps count from the first frame after it. The denoising chunk is the maximum number of future frames sampled jointly.

Condition	Clips	Scored prediction steps	Denoising chunk	DDIM steps	Sampling seeds
In distribution	2,000	1–9	8	10	42, 43, 44
Out of distribution	479	1–29	1	10	42, 43, 44
Sampling-step sweep	512	1	1	1, 2, 4, 6, 8, 10	42
Long rollouts	716	1–59	8	10	42

Metrics. Pixel mean squared error is computed per frame over channels and pixels in $[0, 1]$. SSIM uses a Gaussian window of 11×11 pixels with standard deviation 1.5. LPIPS uses VGG16 features with the learned linear layers of Zhang et al. (2018), in the implementation of Esser et al. (2021), on inputs in $[-1, 1]$; the same network enters the reconstruction loss of every autoencoder with the same weight. For the in- and out-of-distribution evaluations, each framewise metric is averaged over the generated frames of a clip, then over clips, then over the three sampling seeds, which gives one value per trained pipeline; the mean and the standard error are taken over training seeds. The sampling-step sweep and long rollouts use one sampling seed and the frame windows specified below. Out of distribution, the equal-task mean averages the four task means. The latent error of a rollout is the mean squared difference, in the standardized latent space, between a generated latent and the encoder's latent of the recorded frame. FID (Heusel et al., 2017) uses the 2,048-dimensional pool features of Inception-v3 on the nine generated frames of the 2,000 clips, 18,000 frames on each side. FVD (Unterthiner et al., 2018) uses the 400-dimensional outputs of an I3D network (Carreira & Zisserman, 2017) trained on Kinetics-400, with the weights released by Song et al. (2025), on

the 2,000 clips of nine frames. The Inception Score (Salimans et al., 2016) is computed from the softmax of the same I3D outputs over the 400 action classes, as $\exp\left(\mathbb{E}_x \, \mathrm{KL}(p(y \mid x) \,\|\, p(y))\right)$. The three distribution metrics are computed in distribution only, on the same 2,000 clips for each training and sampling seed. We first average the three sampling-seed values of each trained pipeline, then report the mean and standard error over its training seeds.

Long rollouts and sampling steps. In the long-rollout evaluation, the value of a metric at a prediction step is the mean over the videos that extend to that step, so that every model is scored on the same videos at a step; their number falls from 716 at steps 1 to 19 to 224 at steps 57 to 59. A long rollout is summarized by the mean of this value over prediction steps 51 to 59, which 229 to 224 videos reach. The sampling-step sweep generates the first frame of each of the 512 clips with 2, 4, 6, 8 and 10 DDIM steps under the same noise; the matched-budget comparison also includes one DDIM step under the same protocol (Figures 6 and 7). The loss from 10 to 2 steps is the relative increase of the mean error of the first generated frame, computed per training seed and averaged over seeds.

Unpredictable share. We fit the same linear predictor to every frozen autoencoder, including those at $\lambda = 0$, on the 2,000 in-distribution clips (18,000 consecutive-frame transitions). Latents are standardized per channel using 100 batches of training frames, as in Appendix D. Clips are assigned to five folds by their index modulo five in the fixed evaluation order, keeping every transition of a clip in one fold. For each held-out fold, let μ be the mean latent over all frames of the other four folds and let $u_t = H(z_t - \mu)$, where H is the orthonormal two-dimensional Hartley transform applied independently to each channel. We predict each frequency coordinate as

$$\hat{u}_{t+1,j} = b_j u_{t,j} + w_j^\top \phi_t, \tag{7}$$

where ϕ_t concatenates the 256 leading principal coordinates of the centered current latent and the action. The principal directions are fitted on the four training folds; the regression features are standardized using the regression's training data. The coefficients b_j and w_j are fitted jointly by least squares, with ridge penalty $\alpha N \sum_j \|w_j\|_2^2$ for N training transitions; b_j is unpenalized. We select $\alpha \in \{10^{-4}, 10^{-2}, 1, 10^2, 10^4\}$ by fitting on three of the training folds and validating on the fourth (the fold following the outer held-out fold, cyclically), then refit on all four. The reported share is

$$\frac{\sum_{(t,f)} \|z_{t+1} - \hat{z}_{t+1}^{(-f)}\|_2^2}{\sum_{(t,f)} \|z_{t+1} - \mu^{(-f)}\|_2^2}, \tag{8}$$

where the sums contain all held-out transitions and $(-f)$ denotes a fit excluding their fold. Thus the denominator measures the variance of the target latents around the training-fold mean, rather than the variance of their frame-to-frame differences.

Latent forecastability. From one context frame and the recorded action, the model predicts the next latent with 10 DDIM steps and no guidance; the error is the mean squared difference to the encoder's latent of the recorded next frame, divided by the pooled variance of the target latents. It is computed on the first transition of 256 in-distribution clips and on one transition of 256 out-of-distribution trajectories, 64 per task.

Training-curve evaluation. The training-speed factors in Section 5.3 use the validation curves of the reconstruction baselines and temporal objectives, averaged over their three training seeds. These runs validate every 3,318 updates on 400 fixed held-out clips, generating nine frames from one context frame with one frame per denoising chunk, 10 DDIM steps and guidance scale 3; the context frame is excluded from the metrics. For each metric, the reference value is the ordinary-decoder reconstruction baseline's final mean validation value, at 99,540 updates. We find the first crossing of this value by the other mean curve, interpolate linearly in log(update) between its adjacent validation points, and divide 99,540 by the crossing update.

Controls. The reconstruction ceiling encodes and decodes the recorded frames of every evaluation clip with each autoencoder, through the same decoding call as the rollouts, and applies the same metrics; the gap of a world model is its metric minus the ceiling of its autoencoder. For past-conditioned decoders, both use the initial real context frame. The control evaluations use sampling seed 42. The shuffled-action control repeats every rollout with the action sequence of another clip,

assigned by a permutation without fixed points drawn with seed 0 and shared by every model; out of distribution the donor belongs to the same task. The random state is restored before the shuffled rollout, so that it uses the same diffusion noise as the rollout under the recorded actions.

Statistics. The matched-reconstruction experiment uses training seeds 0, 1 and 2, the matched-budget experiment 1234, 2345 and 3456, and the trade-off experiment 1234, with 2345 at weights 0 and 0.3; the seed also fixes the order of the training data, which is therefore shared by the objectives within a seed. The trained pipeline, autoencoder and world model, is the unit of replication; the sampling seeds and the evaluation clips are repeated measurements of one pipeline. We report the mean and the standard error over training seeds and, for each comparison between objectives, the number of seeds in which the change has the better sign.

F MATCHED-RECONSTRUCTION EXPERIMENT

This appendix gives the protocol of the experiment of Section 5.1.

Pipelines. The four pipelines differ only in the objective used to fine-tune the autoencoder; the data, the decoder, the world model and its training, and the evaluation are shared. The decoder is initialized from FLUX.1-schnell and fine-tuned to reconstruct each frame from its latent alone. The temporal weight of Equation 1 is chosen per objective so that fine-tuning reaches the reconstruction target: $\lambda = 0.3$ for latent and contrastive prediction and $\lambda = 0.5$ for pixel prediction.

Stopping rule. Fine-tuning stops at the first end of epoch at which the reconstruction loss on the held-out pairs is at most 0.0345 (Appendix D). Each update count of Table 1 therefore exceeds the number of updates needed to reach the target by less than one epoch of 2,215 updates; with reconstruction alone, the target is met at the second check. For training seeds 0, 1 and 2, fine-tuning stopped after 4,430 updates for every seed with reconstruction alone, 146,190, 152,835 and 170,555 with latent prediction, 8,860 for every seed with pixel prediction, and 97,460, 95,245 and 95,245 with contrastive prediction. The held-out reconstruction loss at the stop is 0.0342 for reconstruction alone and 0.0343 to 0.0345 for the temporal objectives. The dimensionality of Table 1 is the number of principal components that explain 95% of the variance of the flattened latents of 1,000 held-out frames from 100 videos.

World model and evaluation. The world model and its training are described in Appendices C and D, and the evaluation conditions of Table 1 in Appendix E; the long-rollout columns give the mean error over prediction steps 51 to 59. Table 4 adds the distribution metrics to Table 1, and Figure 5 shows the error against the number of sampling steps, along long rollouts and during training.

G MATCHED-BUDGET EXPERIMENT

This appendix gives the protocol of the experiment of Section 5.3.

Pipelines. All fine-tuned pipelines share the data and the budget of 80,000 autoencoder updates on consecutive frames. The pretrained-autoencoder control skips this stage. All pipelines share the world model, its 100,000-update budget, and the evaluation (Appendices C to E). Latent prediction and inverse dynamics use $\lambda = 0.7$ in Equation 1 and the past-conditioned decoder; Appendix H.9 varies the weight. Compared with the matched-reconstruction experiment, the world model is smaller (width 384 and 12 layers, against 1,024 and 16) and trained for fewer updates (100,000, against 238,896), the autoencoder is trained with a larger batch and learning rate, and the latent is standardized per channel rather than globally (Appendix D). The pipelines are trained with seeds 1234, 2345 and 3456.

H ADDITIONAL RESULTS

The additional results test whether the world models use actions, separate the effects of the decoder and the temporal objective, and examine how the gains depend on training and evaluation conditions.

We report the distribution metrics alongside the framewise errors because the two need not improve together.

H.1 CONTROLS

We check whether the generated trajectories depend on the recorded actions, and how much error remains when only the autoencoder is used. The shuffled-action control and the reconstruction ceiling of Appendix E were measured on the reconstruction-only, latent-prediction, pixel-prediction and random-negative contrastive pipelines of the matched-reconstruction experiment (three seeds), and on the reconstruction-only and latent-prediction pipelines with the past-conditioned decoder (three seeds) and inverse dynamics (one seed) of the matched-budget experiment. Under the actions of another clip, the in-distribution pixel error rises by 98 to 102% in the matched-reconstruction experiment and by 62 to 77% in the matched-budget experiment; out of distribution, with the donor clip taken from the same task, it rises by 21 to 26% and 20 to 23%. Encoding and decoding the recorded frames gives a pixel error of 1.9×10^{-4} to 3.1×10^{-4} in the matched-reconstruction experiment and 1.4×10^{-4} to 4.3×10^{-4} in the matched-budget experiment, against 0.011 to 0.034 for the world models on the same autoencoders. The increase under shuffled actions shows that the world models use the action sequence; the lower reconstruction error shows that autoencoding alone does not account for their rollout error.

H.2 COMPLETE QUANTITATIVE RESULTS AND LEARNING CURVES

The main tables summarize framewise prediction quality. Here we add the distribution metrics and show how errors vary with the sampling budget, rollout length and training progress. At each step, all models are evaluated on the same videos; the number of videos decreases with the horizon (Appendix E).

H.2.1 MATCHED RECONSTRUCTION

Table 4 extends the comparison in Section 5.1, with the decoder and reconstruction quality held fixed. Latent prediction gives the largest improvements in perceptual error, with particularly large gains when the sampler has only two steps. The distribution metrics distinguish the objectives: latent prediction gives the lowest FID, whereas contrastive prediction gives the lowest FVD. Figure 5 shows that the differences are smaller along long rollouts and near the end of world-model training than with few sampling steps.

Table 4: Complete results at matched reconstruction quality, including frame-distribution (FID), video-distribution (FVD) and Inception Score (IS) metrics. All pipelines use the ordinary decoder. Reference rows show absolute means and standard errors over three training seeds; pixel MSE is unscaled, in $[0, 1]$ pixel units. Other rows show the mean percentage change against the reference paired by training seed; fractions count seeds that improve. Green/orange shading marks improvement/worsening; arrows give preferred directions; bold marks the best value in each column. Latent prediction improves the framewise metrics most consistently, especially with two sampling steps, while contrastive prediction gives the lowest FVD.

	In distribution			Out of distribution		
	pixel MSE ↓	LPIPS ↓	SSIM ↑	pixel MSE ↓	LPIPS ↓	SSIM ↑
reconstruction only (reference)	0.01165 ±0.00020	0.164 ±0.001	0.801 ±0.002	0.03387 ±0.00128	0.308 ±0.008	0.675 ±0.009
latent future prediction	−3.0% 3/3	**−7.1%** 3/3	**+1.5%** 3/3	**−13.8%** 3/3	**−3.1%** 2/3	**+1.9%** 2/3
pixel future prediction	−1.8% 1/3	−3.3% 3/3	+0.6% 3/3	+1.5% 2/3	−2.4% 2/3	+1.3% 2/3
contrastive future prediction	**−4.1%** 3/3	−2.6% 3/3	+0.7% 3/3	−7.5% 3/3	−1.4% 2/3	+0.9% 2/3

	2 sampling steps			Long rollout, steps 51–59		
	pixel MSE ↓	LPIPS ↓	SSIM ↑	pixel MSE ↓	LPIPS ↓	SSIM ↑
reconstruction only (reference)	0.01595 ±0.00103	0.214 ±0.004	0.770 ±0.005	0.03326 ±0.00184	0.287 ±0.008	0.691 ±0.010
latent future prediction	**−21.9%** 3/3	**−25.2%** 3/3	**+5.0%** 3/3	**−9.1%** 3/3	**−8.4%** 3/3	**+3.7%** 3/3
pixel future prediction	−7.7% 2/3	−7.1% 3/3	+1.6% 3/3	−1.6% 1/3	−2.7% 3/3	+1.0% 2/3
contrastive future prediction	−21.7% 3/3	−13.7% 3/3	+3.2% 3/3	−5.5% 3/3	−3.6% 3/3	+1.4% 3/3

	Distribution metrics		
	FID ↓	FVD ↓	IS ↑
reconstruction only (reference)	9.4 ±0.2	22.9 ±0.4	5.06 ±0.01
latent future prediction	**−11.7%** 3/3	−1.0% 2/3	−1.5% 0/3
pixel future prediction	−1.1% 1/3	+1.3% 1/3	−0.6% 1/3
contrastive future prediction	−2.4% 2/3	**−3.9%** 2/3	**+0.4%** 2/3

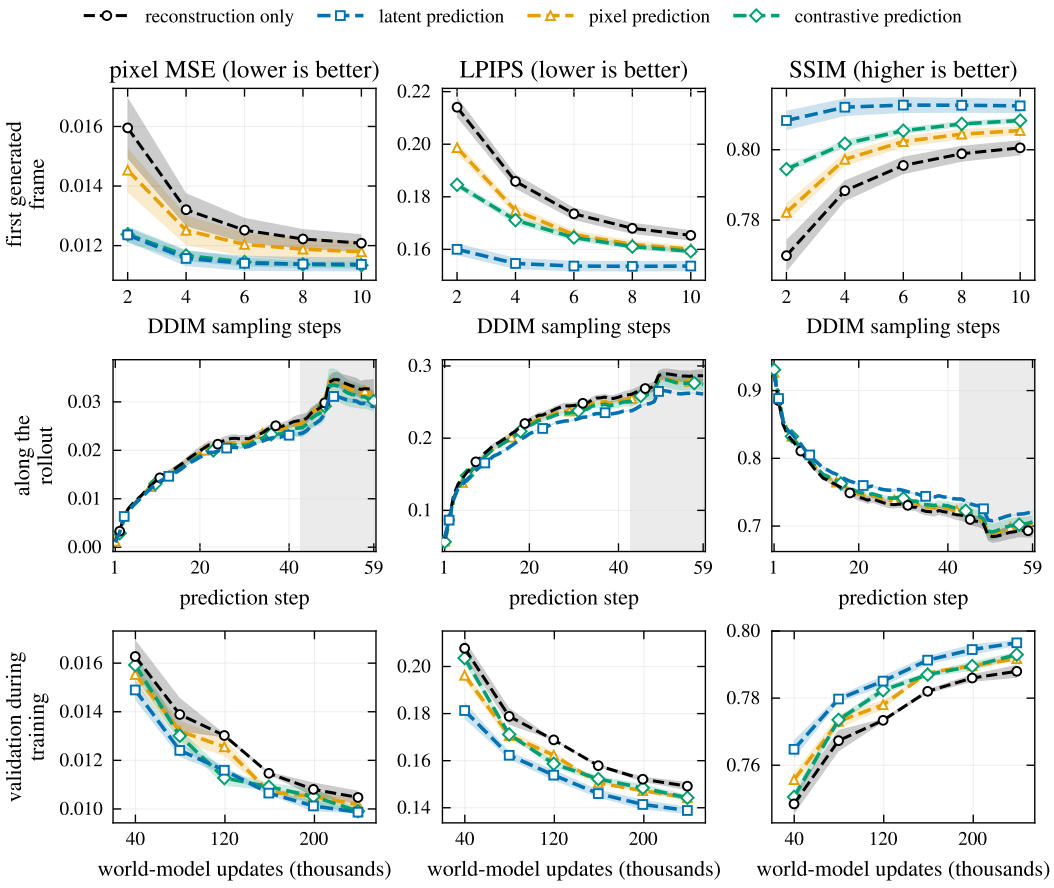


Figure 5: World models on autoencoders trained to equal reconstruction quality, all with the ordinary decoder. Top: first-frame error against DDIM sampling steps on 512 held-out clips. Middle: error at each prediction step on up to 716 held-out videos; grey shading marks steps reached by fewer than half the videos. Bottom: validation during world-model training, every 39,816 updates up to 238,896. Lines and bands show the mean and standard error over three training seeds. Latent prediction gives the largest perceptual gains with few sampling steps; differences are smaller along long rollouts and late in training.

H.2.2 MATCHED TRAINING BUDGET

Table 5 measures the combined benefit of the architecture and objective against reconstruction alone with the ordinary decoder. Table 6 holds the past-conditioned decoder fixed to isolate the objective's contribution.

Table 5: Complete results at matched training budget: 80,000 updates for fine-tuned autoencoders, no updates for the pretrained control, and 100,000 world-model updates. Reference rows show reconstruction alone with the ordinary decoder, as absolute means and standard errors over three training seeds; pixel MSE is unscaled. Other rows show mean percentage changes against the reference paired by seed; fractions count improving seeds. Green/orange shading marks improvement/worsening and bold marks the best value in each column. FID, FVD and IS are distribution metrics. Both temporal objectives improve perceptual error in all four conditions; for latent prediction, the weight trades long-rollout accuracy against distributional quality: at weight 0.7 its FVD is higher, which lower weights avoid (Appendix H.9).

	In distribution			Out of distribution		
	pixel MSE ↓	LPIPS ↓	SSIM ↑	pixel MSE ↓	LPIPS ↓	SSIM ↑
ordinary decoder						
reconstruction only (reference)	0.01434 ±0.00016	0.194 ±0.002	0.779 ±0.001	0.03615 ±0.00098	0.302 ±0.003	0.682 ±0.004
pretrained autoencoder, no fine-tuning	+3.3% 0/3	+7.0% 0/3	−2.7% 0/3	−3.3% 2/3	+2.3% 2/3	−1.3% 2/3
past-conditioned decoder						
reconstruction only	−2.3% 2/3	−9.2% 3/3	+3.0% 3/3	−13.0% 3/3	−10.6% 3/3	+5.4% 3/3
latent future prediction	−0.4% 2/3	−13.0% 3/3	+4.0% 3/3	**−23.6%** 3/3	−14.8% 3/3	+8.5% 3/3
inverse dynamics	**−8.8%** 3/3	**−17.8%** 3/3	**+4.4%** 3/3	−22.7% 3/3	**−21.7%** 3/3	**+9.3%** 3/3

	2 sampling steps			Long rollout, steps 51–59		
	pixel MSE ↓	LPIPS ↓	SSIM ↑	pixel MSE ↓	LPIPS ↓	SSIM ↑
ordinary decoder						
reconstruction only (reference)	0.01835 ±0.00065	0.232 ±0.004	0.756 ±0.003	0.04793 ±0.00151	0.394 ±0.002	0.595 ±0.004
pretrained autoencoder, no fine-tuning	+4.4% 1/3	+7.8% 0/3	−2.9% 0/3	−2.1% 1/3	−0.2% 1/3	−0.9% 1/3
past-conditioned decoder						
reconstruction only	−7.4% 3/3	−14.6% 3/3	+4.3% 3/3	−6.8% 3/3	−7.1% 3/3	+5.1% 3/3
latent future prediction	−14.6% 3/3	−23.7% 3/3	+6.6% 3/3	**−18.4%** 3/3	**−16.0%** 3/3	**+11.3%** 3/3
inverse dynamics	**−26.6%** 3/3	**−30.2%** 3/3	**+7.9%** 3/3	−17.9% 3/3	−15.4% 3/3	+9.8% 3/3

	Latent forecast error ↓	
	in distribution	out of distribution
ordinary decoder		
reconstruction only (reference)	0.232 ±0.008	0.186 ±0.005
pretrained autoencoder, no fine-tuning	+9.6% 0/3	+9.5% 0/3
past-conditioned decoder		
reconstruction only	−8.1% 2/3	−6.9% 2/3
latent future prediction	**−80.3%** 3/3	**−73.7%** 3/3
inverse dynamics	−32.9% 3/3	−29.7% 3/3

	Distribution metrics		
	FID ↓	FVD ↓	IS ↑
ordinary decoder			
reconstruction only (reference)	22.2 ±0.6	39.1 ±0.3	5.04 ±0.01
pretrained autoencoder, no fine-tuning	+29.2% 0/3	+9.1% 0/3	**+2.1%** 3/3
past-conditioned decoder			
reconstruction only	−21.0% 3/3	−8.0% 3/3	−1.0% 0/3
latent future prediction	−3.2% 2/3	+16.4% 0/3	−2.8% 0/3
inverse dynamics	**−49.7%** 3/3	**−15.2%** 3/3	−1.8% 0/3

Both temporal objectives add gains in perceptual error, out-of-distribution pixel error, and the few-step and long-rollout evaluations. For latent prediction, the temporal weight sets a trade-off: at weight 0.7, which favours long rollouts (Appendix H.9), its FVD is higher than that of either reference, and its in-distribution pixel MSE is higher than that of reconstruction alone with the same decoder; lower weights avoid the FVD increase.

Table 6: Effect of the temporal objective at matched training budget with the past-conditioned decoder held fixed. Reference rows show reconstruction alone, as absolute means and standard errors over three training seeds; pixel MSE is unscaled. Other rows show mean percentage changes against the reference paired by seed; fractions count improving seeds. Green/orange shading marks improvement/worsening; bold marks the best value in each column. FID, FVD and IS measure distributional quality. The temporal objectives add gains beyond the decoder in perceptual error, out of distribution, with few sampling steps and on long rollouts.

	In distribution			Out of distribution		
	pixel MSE ↓	LPIPS ↓	SSIM ↑	pixel MSE ↓	LPIPS ↓	SSIM ↑
reconstruction only (reference)	0.01400 ±0.00014	0.176 ±0.002	0.803 ±0.003	0.03145 ±0.00056	0.270 ±0.006	0.718 ±0.006
latent future prediction	+2.0% 1/3	−4.2% 3/3	+0.9% 3/3	**−12.3%** 3/3	−4.6% 3/3	+2.9% 3/3
inverse dynamics	**−6.6%** 3/3	**−9.4%** 3/3	**+1.3%** 3/3	−11.2% 3/3	**−12.4%** 3/3	**+3.7%** 3/3

	2 sampling steps			Long rollout, steps 51–59		
	pixel MSE ↓	LPIPS ↓	SSIM ↑	pixel MSE ↓	LPIPS ↓	SSIM ↑
reconstruction only (reference)	0.01699 ±0.00064	0.198 ±0.003	0.788 ±0.003	0.04461 ±0.00113	0.366 ±0.006	0.626 ±0.007
latent future prediction	−7.7% 3/3	−10.6% 3/3	+2.2% 3/3	**−12.4%** 3/3	**−9.6%** 3/3	**+5.9%** 3/3
inverse dynamics	**−20.6%** 3/3	**−18.3%** 3/3	**+3.5%** 3/3	−11.7% 3/3	−9.0% 3/3	+4.4% 3/3

	Latent forecast error ↓	
	in distribution	out of distribution
reconstruction only (reference)	0.212 ±0.008	0.173 ±0.011
latent future prediction	**−78.4%** 3/3	**−71.5%** 3/3
inverse dynamics	−26.5% 3/3	−23.0% 3/3

	Distribution metrics		
	FID ↓	FVD ↓	IS ↑
reconstruction only (reference)	17.5 ±0.9	36.0 ±0.6	**4.99** ±0.02
latent future prediction	+23.8% 0/3	+26.6% 0/3	−1.8% 0/3
inverse dynamics	**−36.1%** 3/3	**−7.8%** 3/3	−0.8% 1/3

H.2.3 Sampling, rollout and training curves

Figures 6 and 7 show the matched-budget models across all sampling budgets and rollout steps, and during training. The temporal models lose less quality when the sampling budget is reduced and retain lower perceptual error along the rollout. They also reach the baseline's final validation LPIPS earlier; Appendix E defines this training-speed comparison. The shared jump near steps 47–49 reflects the number of evaluated videos falling from 322 to 233.

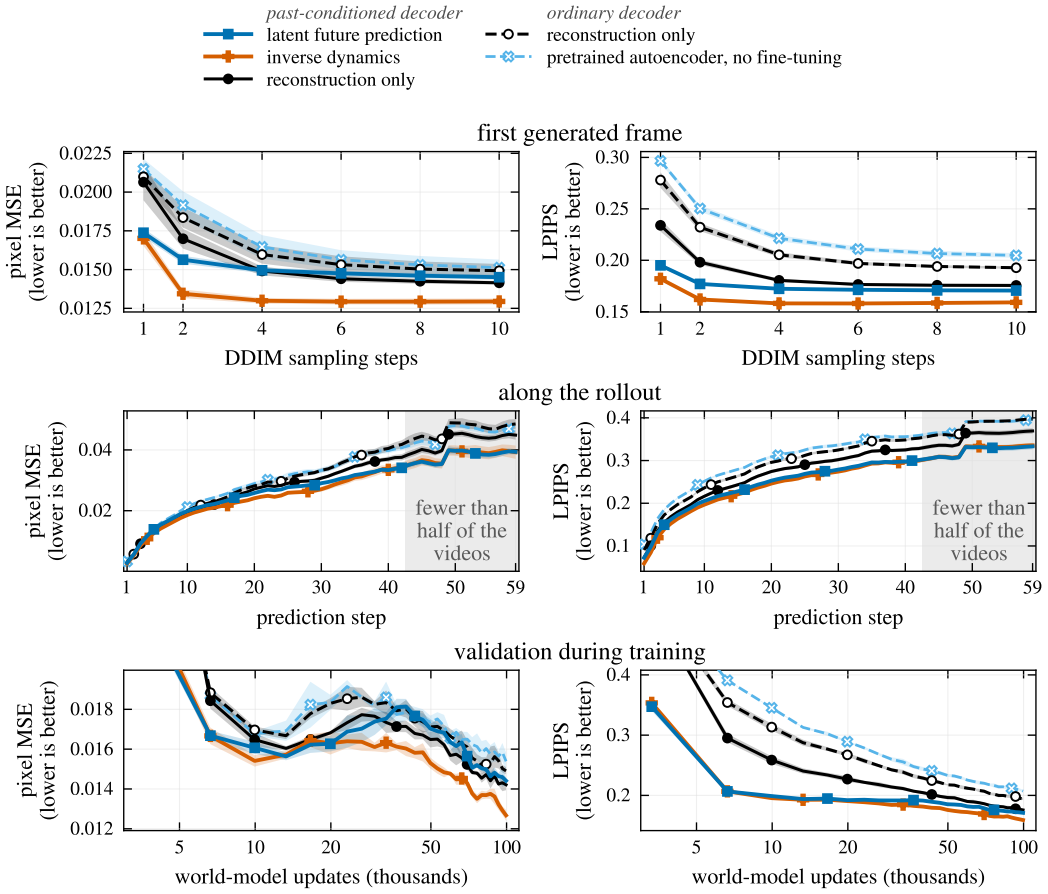


Figure 6: World models trained for 100,000 updates. Autoencoders are fine-tuned for 80,000 updates, except the unchanged pretrained control. Top: first-frame error against DDIM steps on 512 held-out clips. Middle: error at each prediction step on up to 716 held-out videos; grey shading marks steps reached by fewer than half the videos. Bottom: validation during training. Solid lines use the past-conditioned decoder; dashed lines use the ordinary decoder. Mean and standard error over three training seeds. Temporal objectives improve perceptual error across these conditions, with inverse dynamics also giving lower pixel error than reconstruction alone.

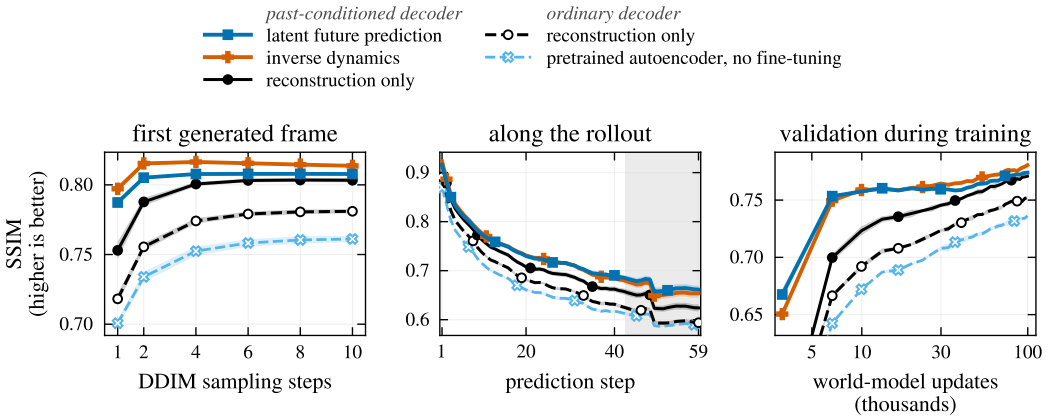


Figure 7: Structural similarity (SSIM; higher is better) for the matched-budget world models. Left: first generated frame on 512 clips against DDIM sampling steps. Centre: each prediction step on up to 716 long videos; grey shading marks steps reached by fewer than half the videos. Right: validation during training. Solid/dashed lines use the past-conditioned/ordinary decoder. Mean and standard error over three training seeds. The past-conditioned decoder improves SSIM, and the temporal objectives add to this gain across the three evaluations.

H.3 Reconstruction and auxiliary prediction losses over training

We track the losses during autoencoder fine-tuning to examine how the reconstruction cost of a temporal objective develops over training (Figure 8). The prediction error here is that of the jointly trained neural predictor. It differs from the unpredictable share in Figure 3, which is measured with a separate linear predictor fitted to each frozen latent space (Appendix E). At a fixed temporal weight, the past-conditioned decoder reaches a lower reconstruction error; increasing the weight lowers the auxiliary prediction error while increasing the reconstruction error.

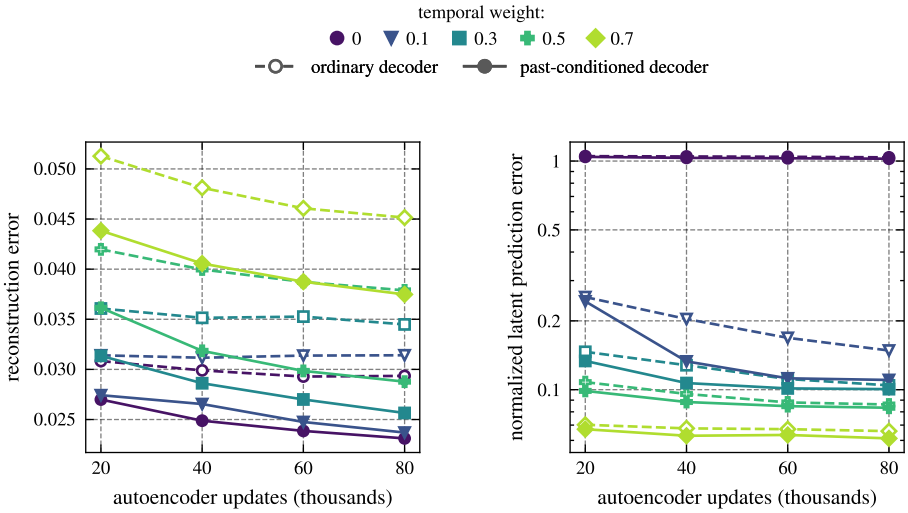


Figure 8: Autoencoder losses during the 80,000-update temporal-weight sweep (training seed 1234). Left: reconstruction error (pixel mean absolute error plus LPIPS). Right: normalized prediction error of the jointly trained neural latent predictor. Colours and marker shapes distinguish temporal weights; open markers and dashed lines use the ordinary decoder, filled markers and solid lines the past-conditioned decoder. The predictor is untrained at weight zero. Past conditioning lowers the reconstruction cost at each weight, while larger temporal weights exchange reconstruction quality for lower auxiliary prediction error.

H.4 Interventions on the past-conditioned decoder

We test whether the improvement from past conditioning depends on the information in the context frame. We evaluate the trained autoencoders with the correct past frame, with the context pathway disabled, and with permuted past frames (Table 7). We randomly permute the past frames and their encoded latents together within evaluation batches of 16, using the same permutation sequence (seed 42) for every checkpoint, while keeping the target latent unchanged. Disabling or replacing the context increases reconstruction error at every temporal weight, with the largest increase at weight 0.7. The same pattern holds for training seed 2345 at weights 0 and 0.3: reconstruction errors are 0.0239 and 0.0261 with the correct past frame, 0.0380 and 0.0514 with the context disabled, and 0.0369 and 0.0498 with permuted past frames.

Table 7: Dependence of the past-conditioned autoencoder on its context after 80,000 updates. Values are reconstruction error (pixel mean absolute error plus LPIPS) on 6,872 held-out frame pairs, training seed 1234; lower is better. Disabling the past input bypasses the context encoder and cross-attention adapters. The last column gives an autoencoder trained with the ordinary decoder at the same temporal weight. Copying the previous frame gives 0.1021. Correct context lowers error at every weight, and dependence on context is greatest at weight 0.7.

weight	past frame	past input disabled	permuted past frame	ordinary decoder
0	0.0231	0.0407	0.0394	0.0294
0.1	0.0237	0.0430	0.0424	0.0314
0.3	0.0257	0.0496	0.0481	0.0345
0.5	0.0288	0.0700	0.0565	0.0379
0.7	0.0375	0.1947	0.1072	0.0451

H.5 POLICY EVALUATION IN WORLDGYM

We test whether the world models preserve differences in policy performance measured on the real robot. WorldGym is a closed-loop evaluation: each policy chooses its next action from the frames generated by its world model, so different world models can induce different action sequences.

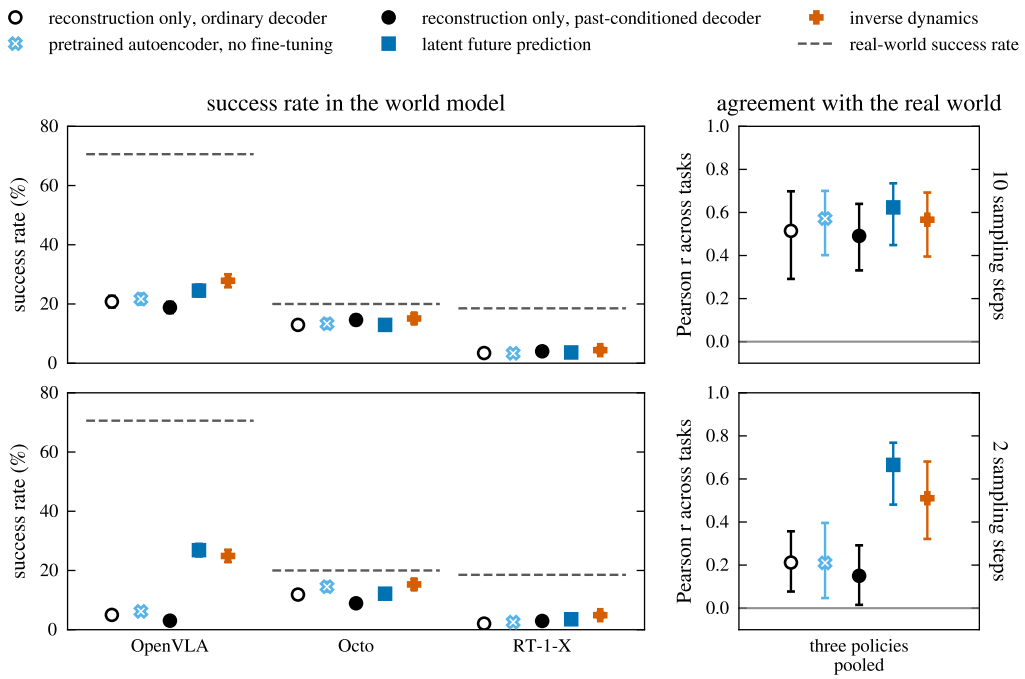


Figure 9: Policy evaluation inside the world models. Three robot policies on the 17 Bridge tasks of WorldGym, inside the five world models of Table 2 (training seed 1234, 100,000 updates); open markers: ordinary decoder, filled markers: past-conditioned decoder. Left: success rate in each world model, mean and standard error across 170 starting frames; dashed line: real-world success rate. Right: Pearson correlation across tasks between the success rates in the world model and in the real world, three policies pooled, with 95% bootstrap interval. Rows: 10 and 2 sampling steps.
With 2 sampling steps, the temporal world models keep their agreement with the real world, while the reconstruction-only world models lose it.

We run the Bridge suite of WorldGym (Quevedo et al., 2026): 17 tasks, compound tasks joined from their two sub-tasks, with 10 starting frames each; from each starting frame, a policy acts for 40 frames inside the world model, three times (once in WorldGym). The world models sample with classifier-free guidance of scale 3 and 10 or 2 sampling steps. The judge is Claude Opus 5.5 at high reasoning effort, in place of WorldGym's GPT-4o, with WorldGym's prompt and, as in WorldGym's released code, frames 0, 20 and 40 of a rollout. It answers 1 for success, 0 for failure, and 0.5 for partial success on the tasks whose criteria define one; it is called three times per rollout (five in WorldGym), the most frequent answer is the outcome, with ties going to the higher score, and a success rate is the mean outcome, so that a partial success counts one half, as in the real-world rates. On 48 real robot episodes of the AutoEval evaluation set with human success labels, 6 successes and 6 failures from each of its four tasks, the majority vote was correct on 42: on all 24 successes and on 18 of the 24 failures. The world models are the five pipelines of Table 2 with training seed 1234, at 100,000 updates. The real-world reference is the per-task success rate of each policy reported by Kim et al. (2025), as used by WorldGym.

Uncertainty. The bootstrap resamples the 17 tasks with replacement, 10,000 times, keeping the three policies of a task together. The standard error of a correlation is the standard deviation of its resampled values, and its interval is the 95% bias-corrected and accelerated interval. The difference between two world models is computed on the same resampled tasks, and its p-value is two-sided.

These quantities describe the sampling of the tasks for one trained world model; unlike the standard errors of the other evaluations, they do not include the variability between training seeds. Table 8 gives the agreement per policy, and, for the pooled agreement, the difference against reconstruction only with the past-conditioned decoder. The world models continued to 200,000 updates are evaluated in Appendix H.7.

With 10 sampling steps, latent prediction has the highest pooled correlation with real-world success. With 2 steps, both temporal objectives retain substantially higher pooled agreement than the reconstruction-only models (Figure 9 and Table 8). The within-policy correlations change less: the largest change is in the comparison between policies, because OpenVLA loses most of its simulated successes in the reconstruction-only models.

Table 8: Policy evaluation in the five world models at 100,000 updates, training seed 1234. Pearson r and Spearman ρ compare simulated and real success rates over 17 tasks per policy, or 51 policy–task pairs when pooled; higher is better. The last three models use the past-conditioned decoder. Pooled r includes the bootstrap standard error over tasks. Δr and p give the paired difference from past-conditioned reconstruction alone and its two-sided bootstrap p-value; – denotes the reference. With two sampling steps, temporal objectives retain higher pooled agreement with real-world performance than all reconstruction-only models.

	OpenVLA		Octo		RT-1-X		three policies pooled			
	r	ρ	r	ρ	r	ρ	r	ρ	Δr	p
10 sampling steps										
reconstruction only, ordinary decoder	0.40	0.27	0.58	0.53	0.11	0.19	0.51 $\pm$0.10	0.45	+0.02	0.623
pretrained autoencoder, no fine-tuning	0.56	0.56	0.58	0.55	0.01	0.10	0.57 $\pm$0.07	0.55	+0.08	0.053
reconstruction only	0.53	0.58	0.50	0.46	0.24	0.49	0.49 $\pm$0.08	0.55	–	–
latent future prediction	0.54	0.53	0.68	0.66	0.08	0.15	0.62 $\pm$0.07	0.64	+0.13	0.011
inverse dynamics	0.40	0.37	0.60	0.58	0.06	0.21	0.57 $\pm$0.07	0.56	+0.08	0.100
2 sampling steps										
reconstruction only, ordinary decoder	0.43	0.36	0.57	0.54	0.21	0.28	0.21 $\pm$0.07	0.30	+0.06	0.007
pretrained autoencoder, no fine-tuning	0.47	0.43	0.57	0.58	0.13	0.36	0.21 $\pm$0.09	0.30	+0.06	0.210
reconstruction only	0.40	0.42	0.54	0.54	0.30	0.45	0.15 $\pm$0.07	0.28	–	–
latent future prediction	0.55	0.49	0.57	0.54	0.56	0.63	0.67 $\pm$0.07	0.69	+0.52	$<$0.001
inverse dynamics	0.36	0.35	0.57	0.44	0.02	0.18	0.51 $\pm$0.09	0.47	+0.36	$<$0.001

H.6 Qualitative examples

Rollouts. Figures 11 to 18 show rollouts of the five world models of Table 2 (training seed 1234, 100,000 updates) in two evaluation conditions of Appendix E, each with 2 and with 10 sampling steps. Out of distribution, the clips are the 479 AutoEval trajectories, and each world model replays the recorded actions for 29 frames from the context frame, one frame at a time. In distribution, the clips are the 224 held-out Bridge videos with at least 60 frames of the long-rollout evaluation, and each world model rolls out 59 frames from the first frame, eight at a time. All world models roll out the same clip from the same context frame, with the same actions and the same diffusion noise; the noise seeds differ from those of the quantitative evaluation, so the LPIPS under each row is that of the rollout shown. For each condition, one figure shows two selected clips and one figure shows two clips drawn at random.

All clips. Table 9 places the examples in the distribution over all clips. Both temporal world models have a lower LPIPS than the baseline on 92.0% to 100% of the clips in every condition, with median reductions of 13.1% to 31.8%. The gap is largest with 2 sampling steps: out of distribution, latent prediction has a lower LPIPS than the baseline on all 479 trajectories, and both temporal world models are below all three reconstruction-only world models on 96.2% of them. With 10 steps, the reconstruction-only world models degrade more slowly, and the differences appear mainly late in the long rollouts.

Policy evaluation. Figure 19 shows OpenVLA acting inside the world models in WorldGym (Appendix H.5) with 2 sampling steps. The loop is closed: the policy acts on the frames that each world model generates, so the rows share the starting frame, the instruction and the policy, but not the

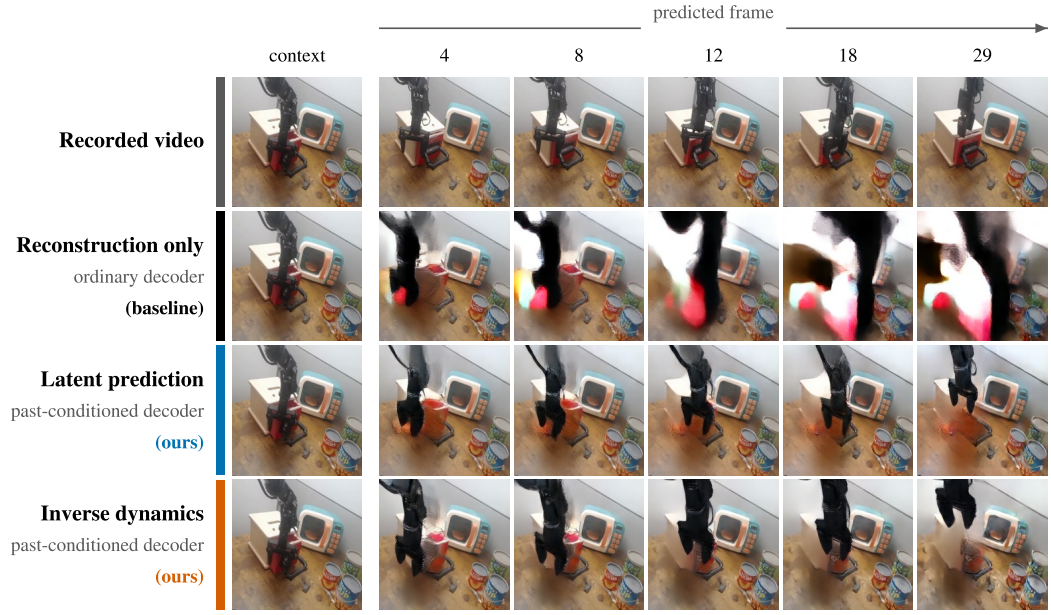
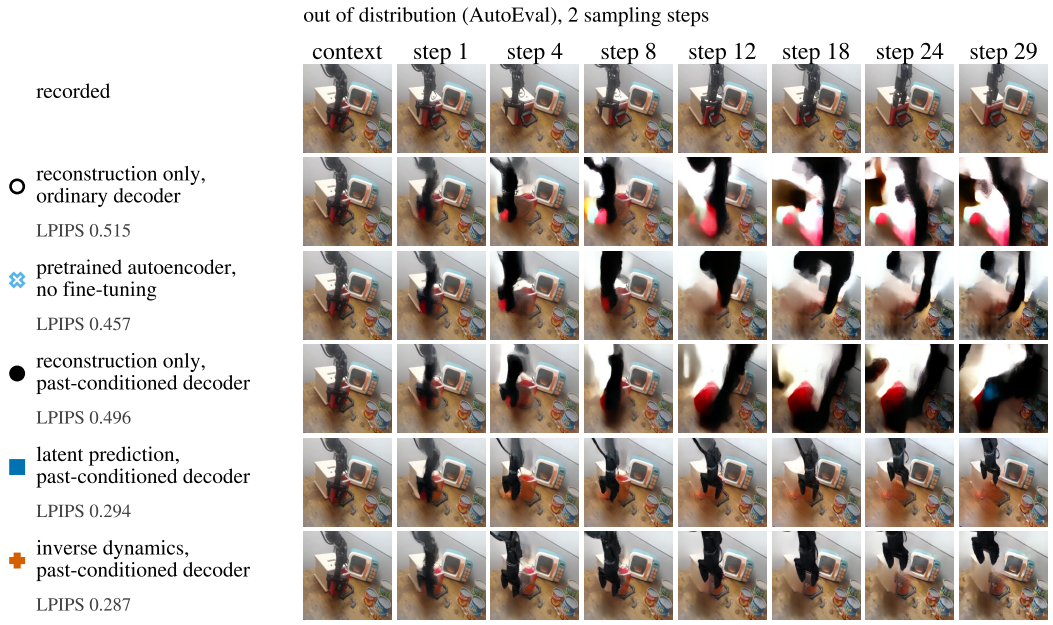


Figure 10: **AutoEval rollouts with video world models trained on different latent spaces.** The baseline follows standard practice and learns the latent space through reconstruction; our recipe adds a temporal objective (latent prediction or inverse dynamics), and a past-conditioned decoder. The rollouts are out of distribution, use only 2 sampling steps, and extrapolate to three times the length of the training clips.
The baseline has strong artifacts, does not follow the ground truth movement, and loses stability over long horizon; our recipes maintain stable generations, higher realism, and better follow the true movement.

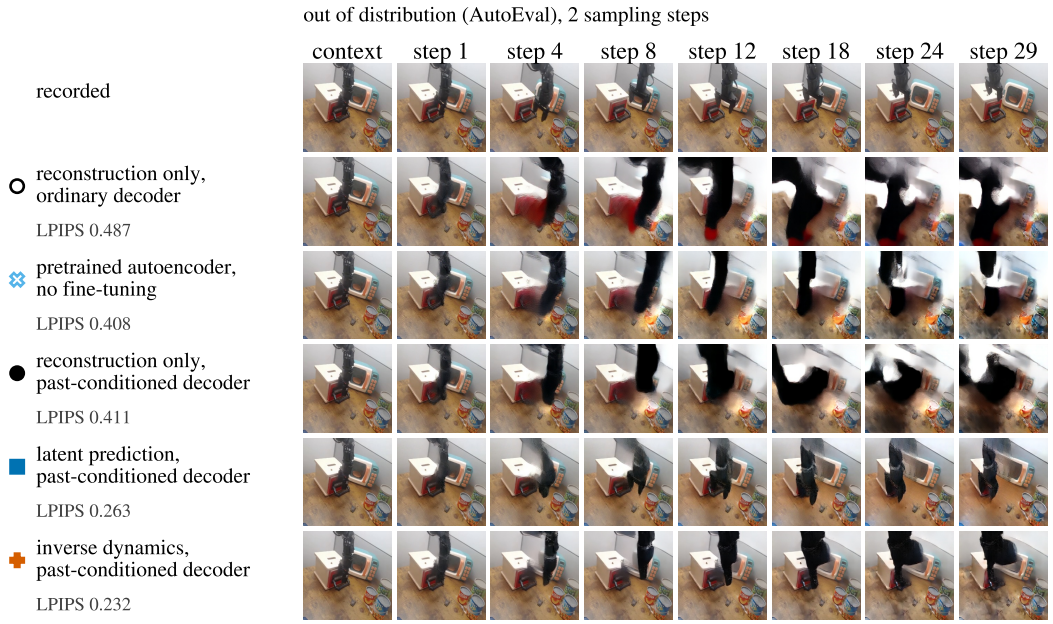
Table 9: Qualitative comparison over all clips of each condition: percentage of clips on which latent prediction (latent) and inverse dynamics (inverse) have a lower clip-mean LPIPS than the baseline (reconstruction only, ordinary decoder), and than all three reconstruction-only world models (the baseline, the pretrained autoencoder, and reconstruction only with the past-conditioned decoder; both: the two temporal world models at once); and the median relative change of LPIPS against the baseline. One rollout per clip.
On almost every clip, the temporal world models are closer to the recording than the baseline, most consistently with 2 sampling steps.

			lower than baseline (%)		lower than all reconstruction-only (%)			median change vs. baseline (%)	
setting	steps	clips	latent	inverse	latent	inverse	both	latent	inverse
in distribution	10	224	92.0	93.3	80.4	79.9	69.6	−13.1	−13.6
in distribution	2	224	96.0	98.7	89.3	92.9	87.5	−21.5	−21.1
out of distribution	10	479	95.0	96.7	60.5	86.2	53.2	−13.9	−22.8
out of distribution	2	479	100.0	99.2	97.9	97.9	96.2	−28.9	−31.8

actions. In 78 of the 510 rollouts, from 53 of the 170 starting frames, both reconstruction-only world models with a fine-tuned autoencoder are judged failures and a temporal world model is judged a success; the reverse, both temporal world models judged failures and a reconstruction-only world model judged a success, does not occur.

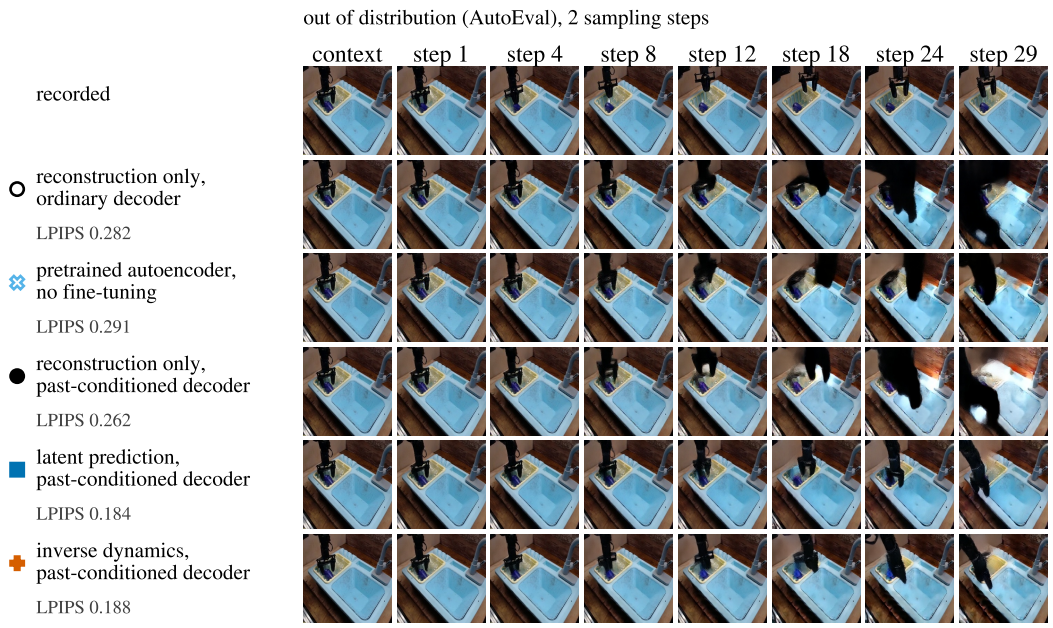


(a) “Close the drawer”.

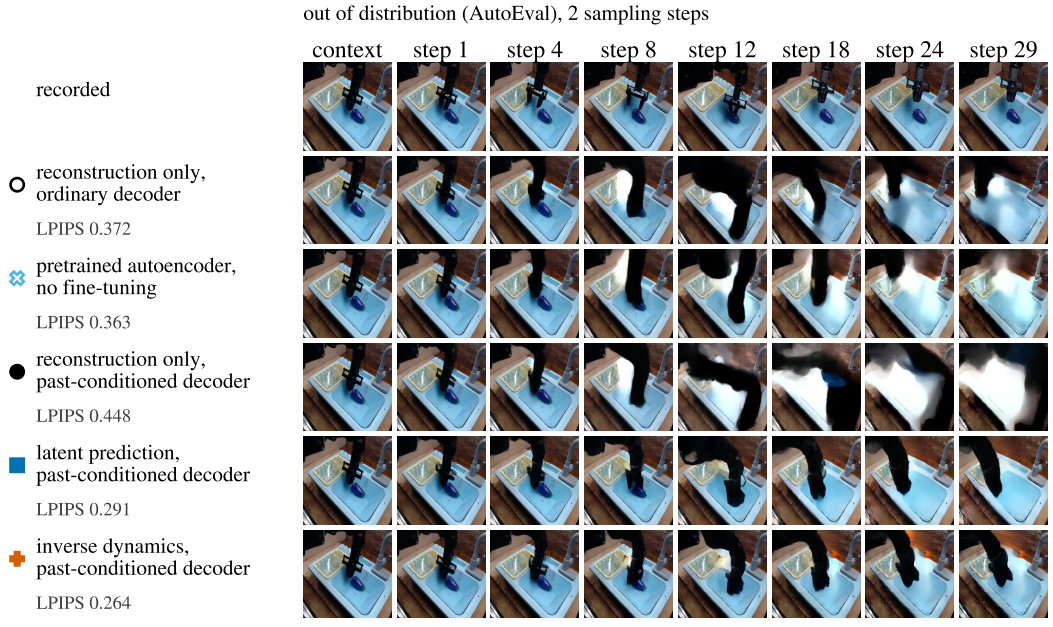


(b) “Close the drawer”.

Figure 11: Out of distribution, 2 sampling steps, selected trajectories: AutoEval trajectories with the recorded actions replayed. Top row: the recorded frames; below: the rollout of each world model from the same context frame, with its LPIPS.
All three reconstruction-only world models, including the one with the past-conditioned decoder, dissolve the arm and the scene within 4 to 12 frames, while both temporal world models keep them and follow the recorded motion.

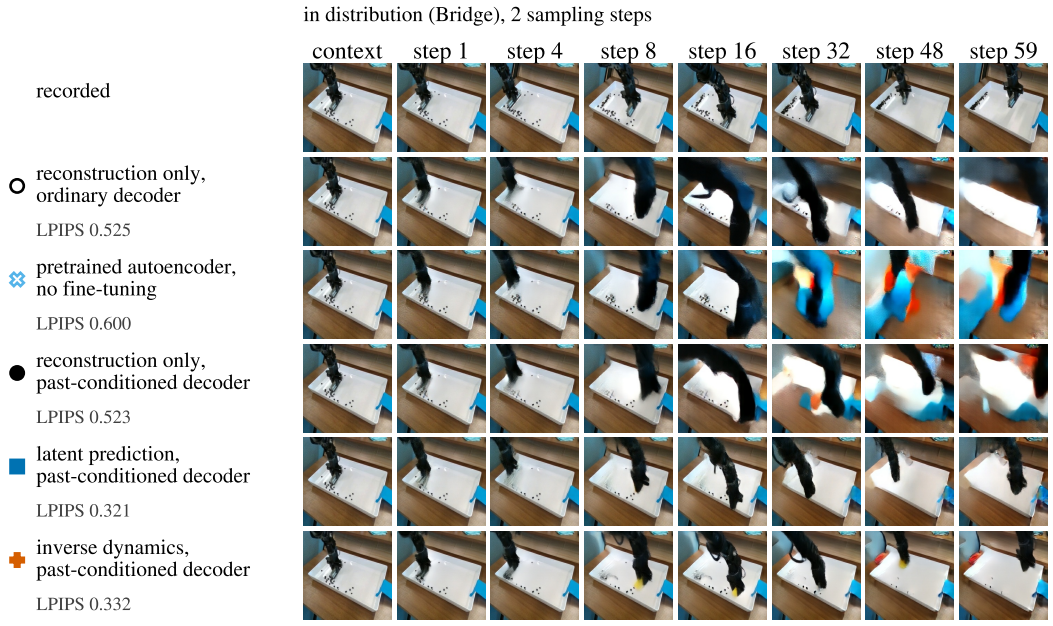


(a) “Put the eggplant in the blue sink”.

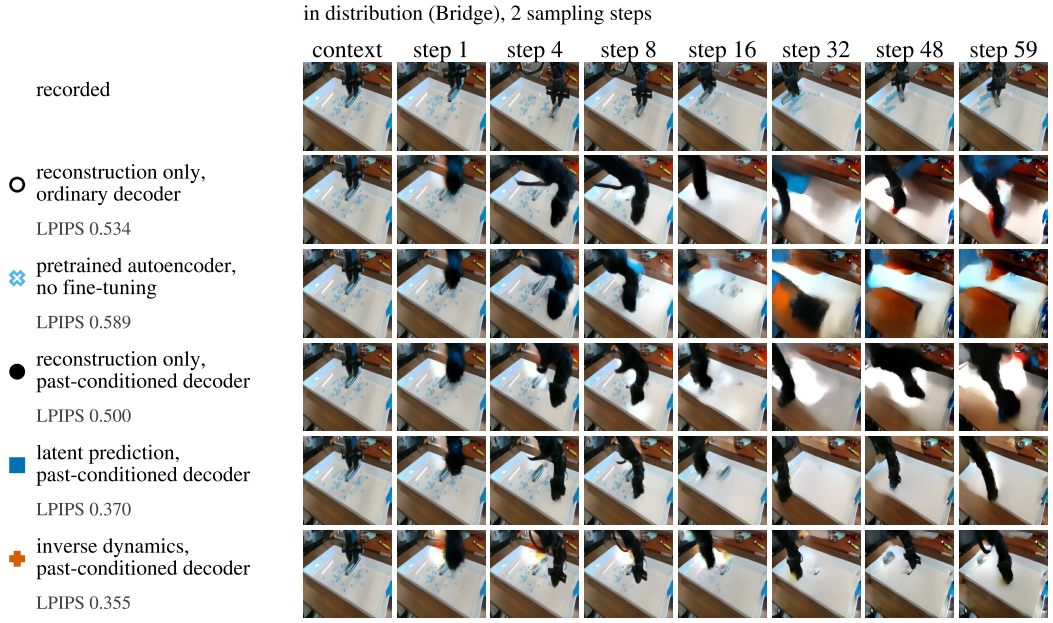


(b) “Put the eggplant in the yellow basket”.

Figure 12: Out of distribution, 2 sampling steps, random trajectories. Rows as in Figure 11. On random trajectories too, the reconstruction-only world models smear or dissolve the arm within 12 to 24 frames, while the temporal world models keep the arm and the scene.

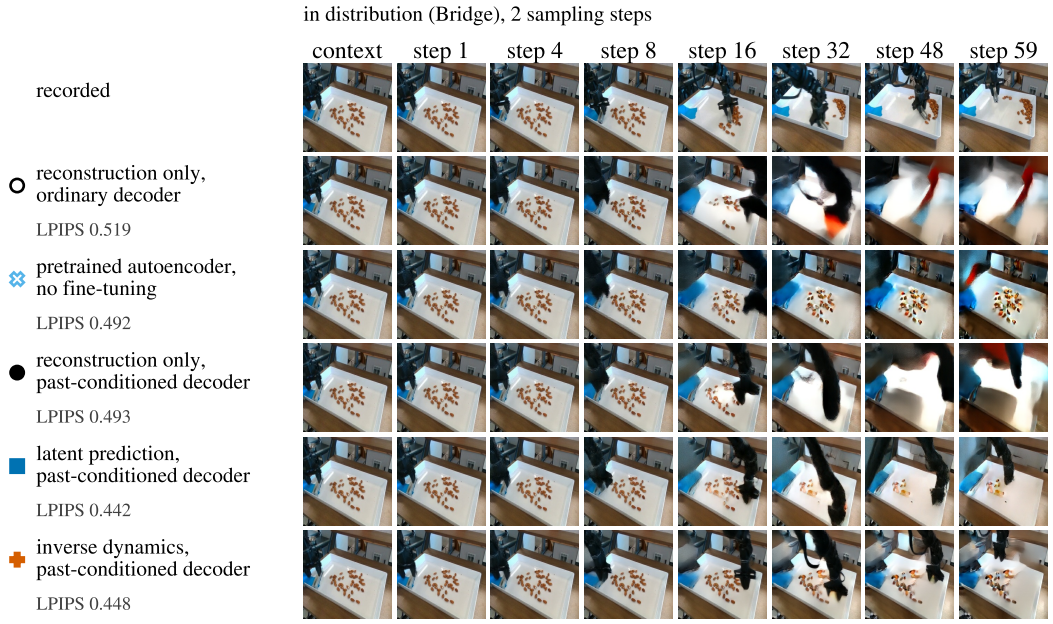


(a)

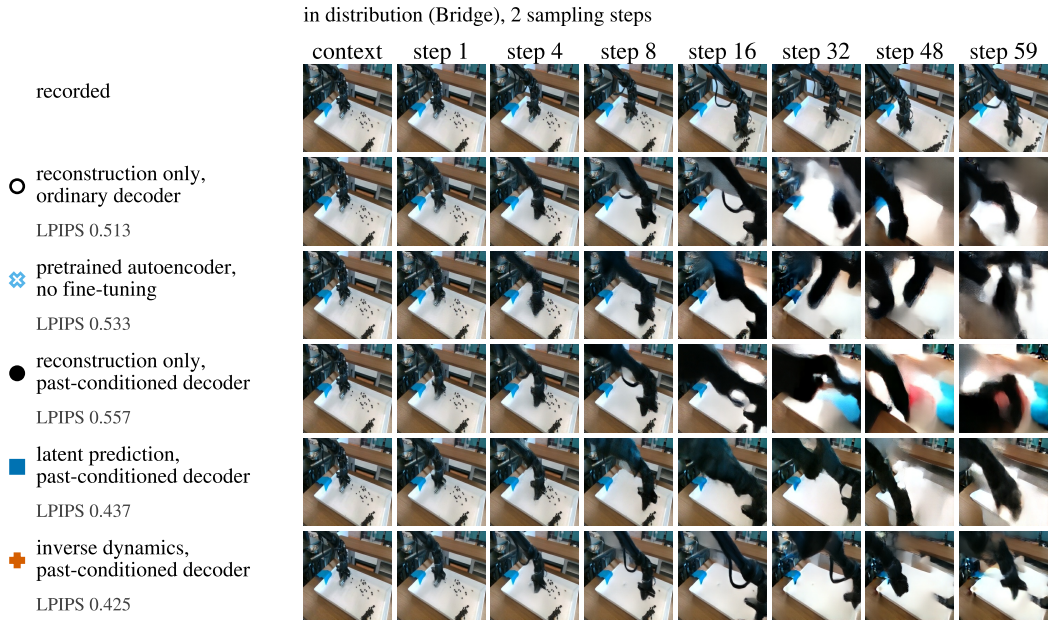


(b)

Figure 13: In distribution, 2 sampling steps, selected videos: held-out Bridge videos rolled out for 59 frames. Rows as in Figure 11.
Within 16 to 32 frames, the reconstruction-only world models replace the scene with saturated blobs, while the temporal world models keep the scene and the arm to the end of the rollout.

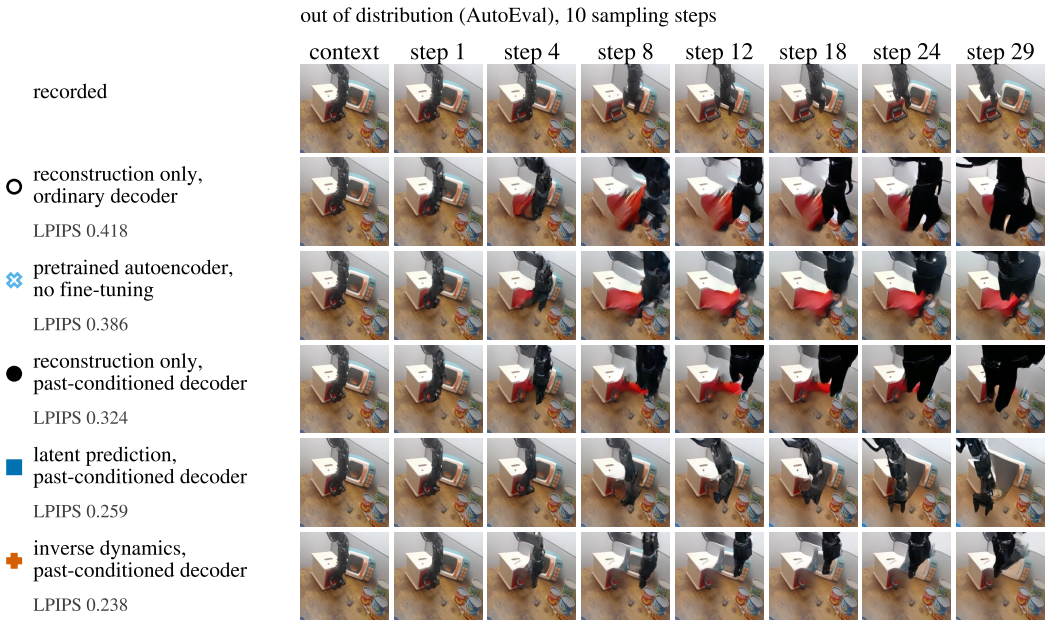


(a)

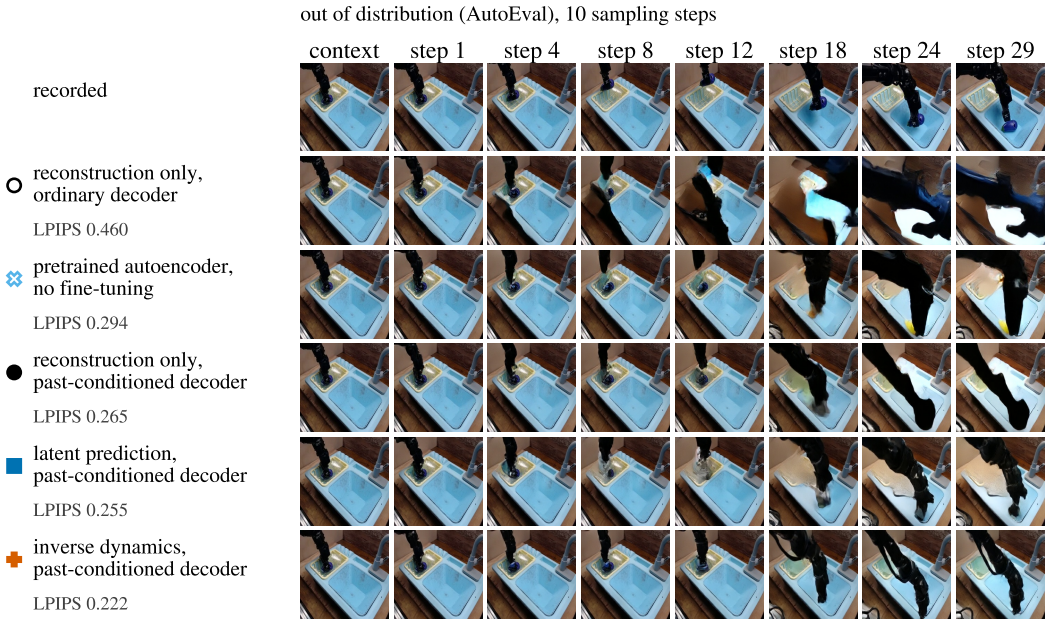


(b)

Figure 14: In distribution, 2 sampling steps, random videos. Rows as in Figure 11.
On random videos too, the reconstruction-only world models wash out the scene within 32 to 48 frames, while the temporal world models keep the arm and the scene.

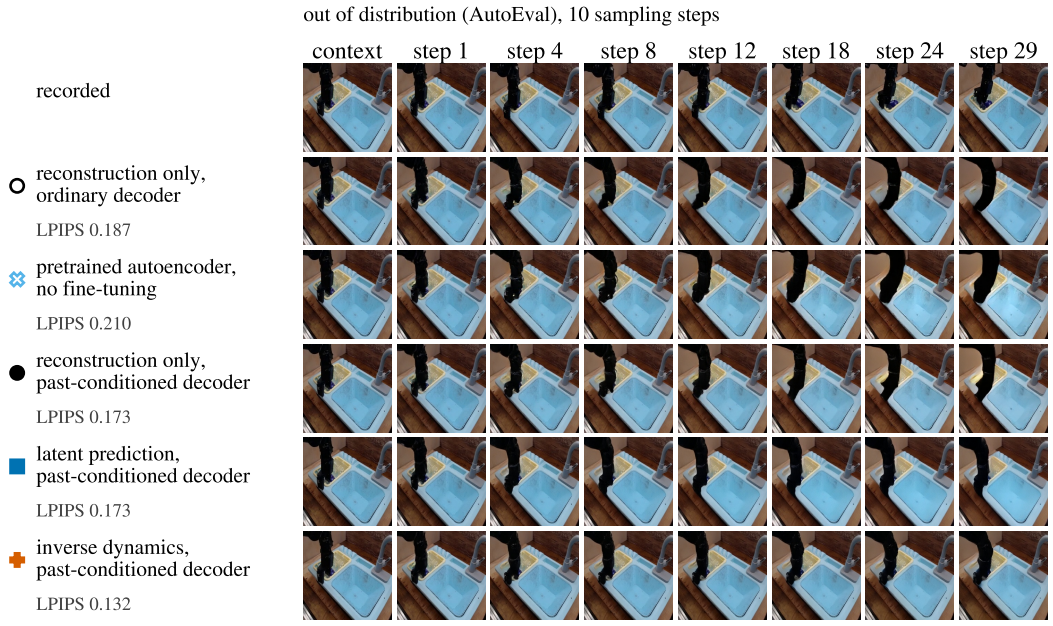


(a) “Close the drawer”.

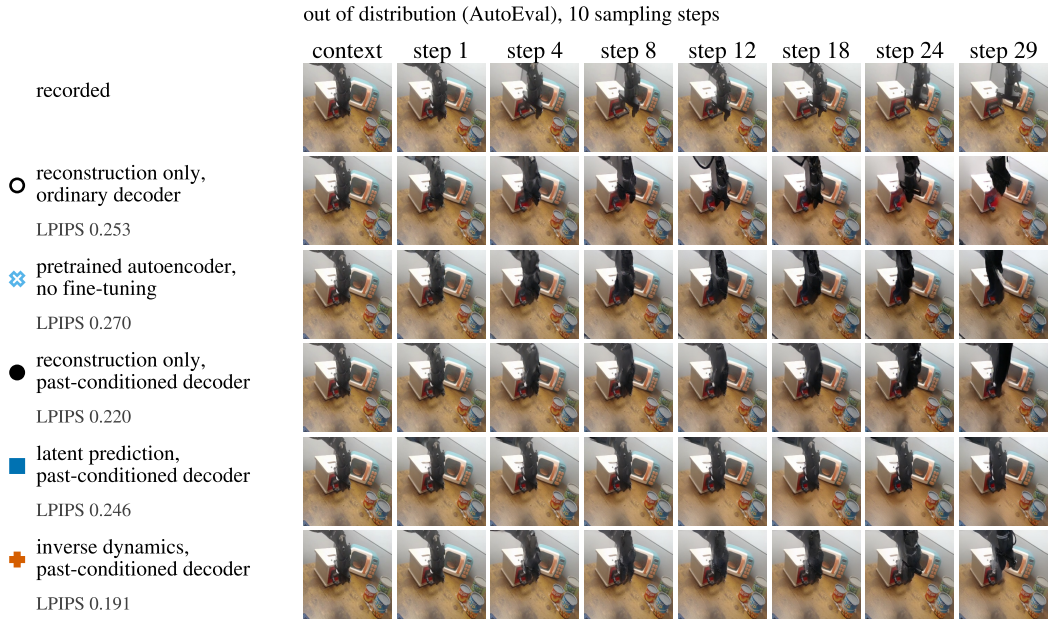


(b) “Put the eggplant in the blue sink”.

Figure 15: Out of distribution, 10 sampling steps, selected trajectories. Rows as in Figure 11. With 10 steps, the reconstruction-only world models degrade later, but still invent a red drawer front and lose the arm (a) or collapse the scene after 18 frames (b), while the temporal world models follow the recorded motion.

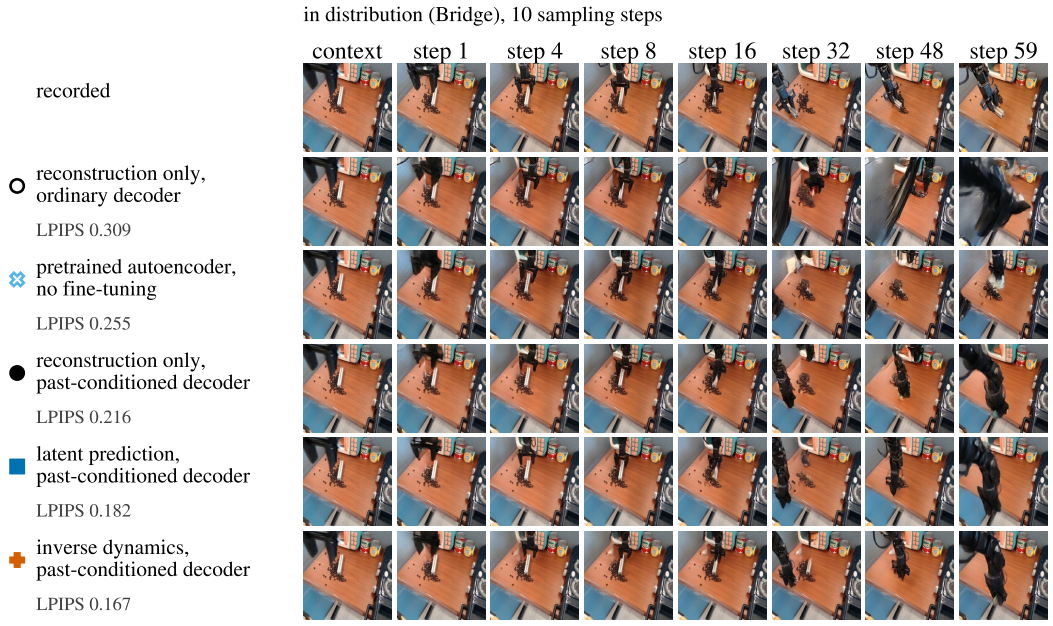


(a) “Put the eggplant in the blue sink”.

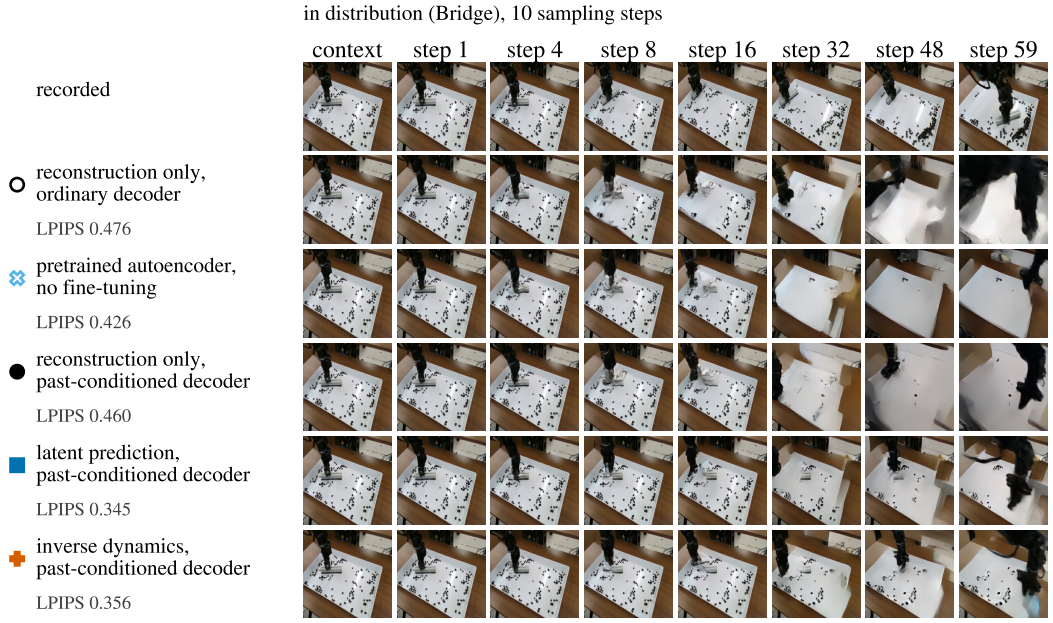


(b) “Open the drawer”.

Figure 16: Out of distribution, 10 sampling steps, random trajectories. Rows as in Figure 11. With 10 steps, every world model keeps the scene on these trajectories, and the differences between them are small.

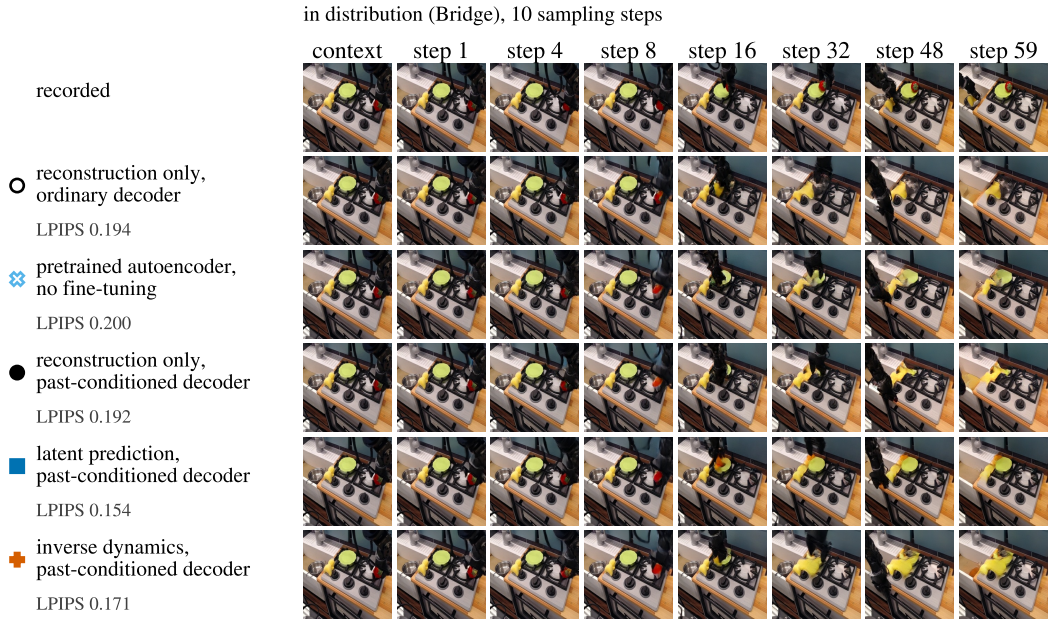


(a)

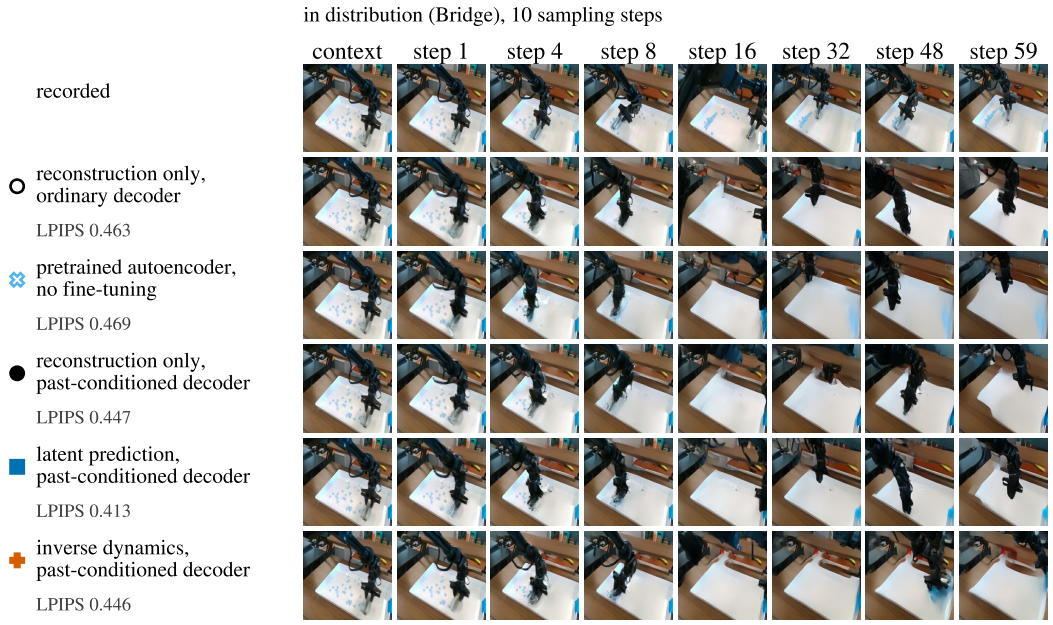


(b)

Figure 17: In distribution, 10 sampling steps, selected videos. Rows as in Figure 11.
With 10 steps, the differences appear late in the rollout: after 32 to 48 frames, the reconstruction-only world models distort the arm or wash out the scene, while the temporal world models stay close to the recording.

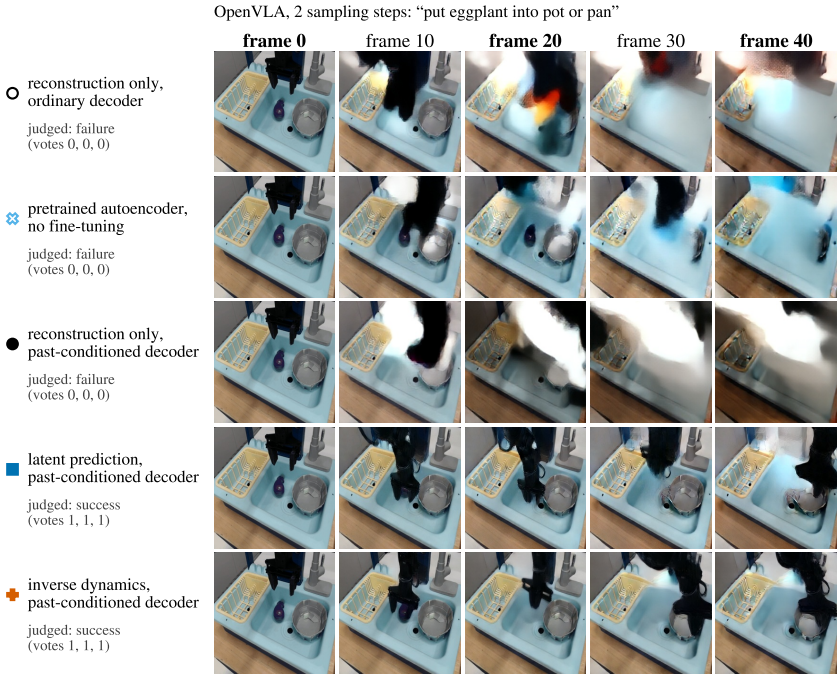


(a)

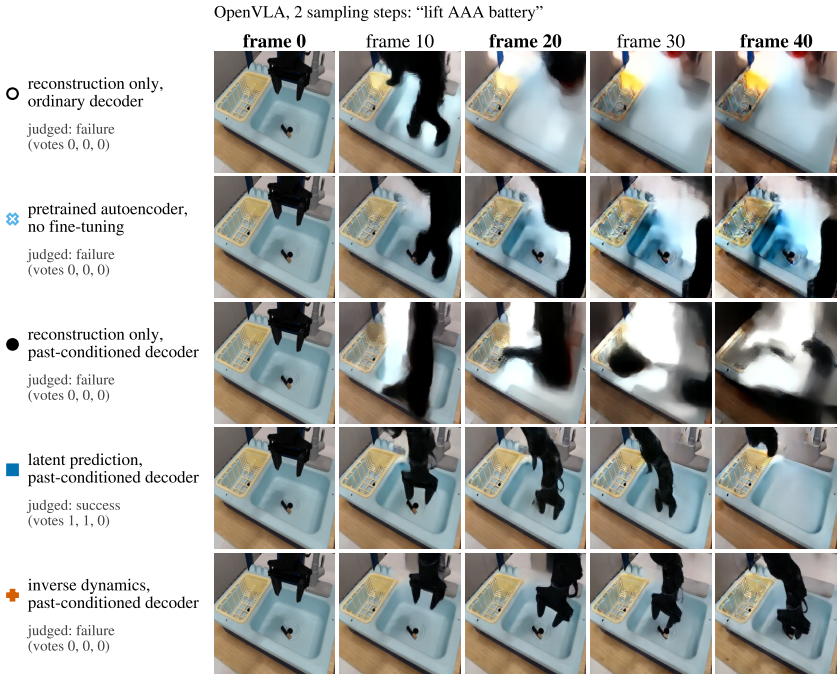


(b)

Figure 18: In distribution, 10 sampling steps, random videos. Rows as in Figure 11.
With 10 steps, every world model keeps the scene on these videos; the temporal world models stay slightly closer to the recording.



(a) Selected.



(b) Random starting frame.

Figure 19: OpenVLA in WorldGym with 2 sampling steps, from one starting frame, inside the world models (training seed 1234, 100,000 updates), at frames 0 to 40; bold: the frames the judge sees. Under each row: the judged outcome and the three votes.
The reconstruction-only world models wash out the scene within 20 to 30 frames, while the temporal world models keep the scene and the arm for the whole episode.

H.7 LONGER WORLD-MODEL TRAINING

The three past-conditioned world models of Section 5.3 with training seeds 1234 and 2345 were continued from 100,000 to 200,000 updates with the same settings. Table 10 compares them at 200,000 updates, and gives the same two seeds at 100,000 updates for comparison. With longer training, inverse dynamics keeps its gains in all four evaluation conditions. For latent prediction, the trade-off shifts with longer training: it extends its gain out of distribution (pixel MSE -24.4%, against -16.0% at 100,000 updates, in both seeds) and keeps a lower LPIPS and a higher SSIM in all four conditions, in both seeds, while its pixel-MSE gain on long rollouts narrows (-5.0%, against -12.5%), its in-distribution pixel MSE rises above that of reconstruction alone ($+5.9\%$), and its FVD remains higher.

Policy evaluation. The world models of training seed 1234 at 200,000 updates were also evaluated in WorldGym, as in Appendix H.5 (Table 11 and Figure 20). The pattern at 100,000 updates holds: with 10 sampling steps both temporal world models agree better with the real world than reconstruction only ($\Delta r = +0.07$ and $+0.05$; $p = 0.086$ and 0.081), and with 2 steps the agreement of reconstruction only collapses (pooled r from 0.61 to 0.20) while that of the temporal world models changes little (from 0.69 to 0.68 and from 0.66 to 0.60).

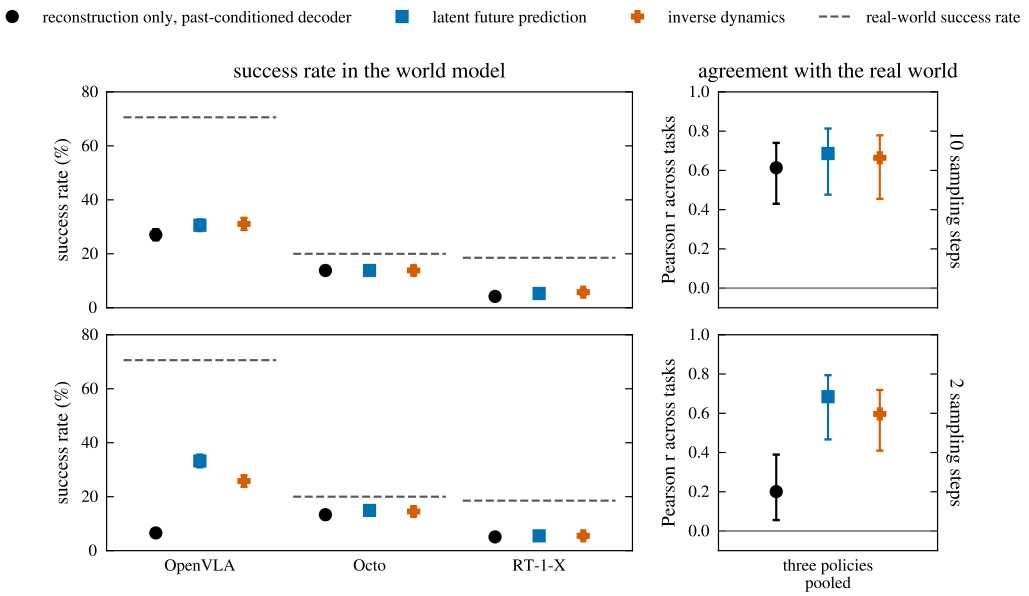


Figure 20: Policy evaluation after 200,000 updates, training seed 1234, for the three past-conditioned world models. Left: simulated success of OpenVLA, Octo and RT-1-X on 17 tasks, mean and standard error over 170 starting frames; dashed lines give real-world success. Right: Pearson correlation of simulated and real success across 51 policy–task pairs, with 95% task-bootstrap intervals. Rows use ten and two sampling steps. The temporal objectives preserve the ordering of policy performance more closely when the sampling budget falls to two steps.

Table 10: Effect of longer world-model training on the three past-conditioned pipelines, with autoencoders fixed. (a) 200,000 updates; (b) the same training seeds, 1234 and 2345, at 100,000 updates. Each reference row gives absolute means and standard errors for reconstruction alone; pixel MSE is unscaled. Other rows show mean percentage changes paired by seed; fractions count improving seeds. Green/orange shading marks improvement/worsening; bold marks the best value. Latent forecast error was measured only at 100,000 updates. Inverse dynamics retains gains in all four framewise evaluations; latent prediction extends its out-of-distribution gain and keeps a lower LPIPS in every condition, while its in-distribution pixel MSE rises above that of reconstruction alone.

	In distribution			Out of distribution		
	pixel MSE ↓	LPIPS ↓	SSIM ↑	pixel MSE ↓	LPIPS ↓	SSIM ↑
(a) 200,000 world-model updates						
reconstruction only (reference)	0.01214 ±0.00036	0.163 ±0.001	0.813 ±0.002	0.03453 ±0.00233	0.280 ±0.006	0.707 ±0.006
latent future prediction	+5.9% 0/2	−3.3% 2/2	+0.6% 2/2	**−24.4%** 2/2	−9.1% 2/2	**+5.1%** 2/2
inverse dynamics	**−4.7%** 2/2	**−8.8%** 2/2	**+1.2%** 2/2	−9.5% 2/2	**−13.6%** 2/2	+4.4% 2/2
(b) 100,000 world-model updates, same training seeds						
reconstruction only (reference)	0.01400 ±0.00024	0.179 ±0.000	0.800 ±0.001	0.03200 ±0.00019	0.276 ±0.002	0.712 ±0.001
latent future prediction	+0.7% 1/2	−5.3% 2/2	+1.2% 2/2	**−16.0%** 2/2	−6.8% 2/2	+4.1% 2/2
inverse dynamics	**−4.9%** 2/2	**−9.9%** 2/2	**+1.5%** 2/2	−10.2% 2/2	**−13.5%** 2/2	**+4.4%** 2/2

	2 sampling steps			Long rollout, steps 51–59		
	pixel MSE ↓	LPIPS ↓	SSIM ↑	pixel MSE ↓	LPIPS ↓	SSIM ↑
(a) 200,000 world-model updates						
reconstruction only (reference)	0.01489 ±0.00064	0.187 ±0.001	0.795 ±0.001	0.03727 ±0.00064	0.321 ±0.000	0.670 ±0.001
latent future prediction	−6.0% 2/2	−12.2% 2/2	+2.4% 2/2	−5.0% 1/2	−7.2% 2/2	+3.4% 2/2
inverse dynamics	**−17.7%** 2/2	**−18.6%** 2/2	**+3.4%** 2/2	**−10.0%** 2/2	**−9.1%** 2/2	**+3.6%** 2/2
(b) 100,000 world-model updates, same training seeds						
reconstruction only (reference)	0.01681 ±0.00106	0.200 ±0.003	0.786 ±0.004	0.04571 ±0.00044	0.372 ±0.001	0.619 ±0.002
latent future prediction	−8.1% 2/2	−11.5% 2/2	+2.5% 2/2	**−12.5%** 2/2	**−9.8%** 2/2	**+6.2%** 2/2
inverse dynamics	**−18.1%** 2/2	**−18.5%** 2/2	**+3.6%** 2/2	−11.5% 2/2	−9.4% 2/2	+4.8% 2/2

	Latent forecast error ↓	
	in distribution	out of distribution
(b) 100,000 world-model updates, same training seeds		
reconstruction only (reference)	0.219 ±0.008	0.182 ±0.010
latent future prediction	**−79.3%** 2/2	**−73.2%** 2/2
inverse dynamics	−29.3% 2/2	−30.8% 2/2

	Distribution metrics		
	FID ↓	FVD ↓	IS ↑
(a) 200,000 world-model updates			
reconstruction only (reference)	12.1 ±0.0	28.4 ±0.7	**4.99** ±0.03
latent future prediction	+17.6% 0/2	+32.8% 0/2	−1.6% 0/2
inverse dynamics	**−26.2%** 2/2	**−0.7%** 1/2	−1.2% 0/2
(b) 100,000 world-model updates, same training seeds			
reconstruction only (reference)	18.3 ±0.2	36.6 ±0.1	**5.01** ±0.00
latent future prediction	+16.2% 0/2	+25.5% 0/2	−2.5% 0/2
inverse dynamics	**−37.7%** 2/2	**−8.5%** 2/2	−1.3% 0/2

Table 11: Policy evaluation after 200,000 world-model updates, training seed 1234; all three models use the past-conditioned decoder. Pearson r and Spearman ρ compare simulated and real success rates on 17 tasks per policy or 51 policy–task pairs pooled. Pooled r includes the task-bootstrap standard error. Δr and p compare with reconstruction alone using paired task resampling; – denotes the reference. At two sampling steps, latent prediction and inverse dynamics retain higher pooled agreement than reconstruction alone; the differences at ten steps are smaller.

	OpenVLA		Octo		RT-1-X		three policies pooled			
	r	ρ	r	ρ	r	ρ	r	ρ	Δr	p
10 sampling steps										
reconstruction only	0.50	0.45	0.55	0.53	0.24	0.41	0.61 $\pm$0.08	0.61	–	–
latent future prediction	0.61	0.58	0.53	0.52	0.56	0.58	0.69 $\pm$0.08	0.68	+0.07	0.086
inverse dynamics	0.54	0.47	0.57	0.53	0.34	0.30	0.66 $\pm$0.07	0.65	+0.05	0.081
2 sampling steps										
reconstruction only	0.44	0.40	0.53	0.53	0.21	0.51	0.20 $\pm$0.08	0.31	–	–
latent future prediction	0.56	0.52	0.52	0.40	0.49	0.47	0.68 $\pm$0.07	0.65	+0.48	$<$0.001
inverse dynamics	0.50	0.46	0.58	0.55	0.55	0.70	0.60 $\pm$0.07	0.61	+0.40	$<$0.001

H.8 LATENT SPACES

To describe what the temporal objectives change in the latent space, we encode 1,000 held-out frames with the frozen encoders of both experiments. We sample ten consecutive frames from each of 100 videos, choosing the start frame uniformly within each video. We measure the dimensionality of the flattened latents (principal components for 95% of the variance, and the participation ratio of the spectrum), the straightness of their trajectories (the mean angle between successive latent displacements along a video, divided by the same angle in pixel space), how closely frames of the same video cluster (the mean distance between frames of one video divided by that between frames of different videos), and the spatial smoothness of the latent maps. For smoothness, we compute the real two-dimensional Fourier transform of each latent channel and average its magnitudes over frames and channels. Let $A(r)$ be the mean amplitude of the stored frequency coordinates at rounded radial frequency r, in cycles per 32 latent positions. We report the least-squares slope of $\log A(r)$ against $\log r$ for $r = 1, \ldots, 16$, and the high-frequency spectral share $\sum_{r=8}^{16} A(r)^2 / \sum_{r=1}^{16} A(r)^2$. The two experiments use different frames, so their values are not compared with each other. Table 12 gives the statistics, Figure 23 the spectra, and Figures 21 and 22 the latent maps of one frame.

Latent prediction changes the latent space in the same direction in both experiments. It reduces the number of principal components needed for 95% variance (177 against 388 at matched reconstruction, and 110 against 316 at matched budget), the participation ratio, the turning-angle ratio, spatial roughness and the within-to-across-video distance ratio. Contrastive prediction shifts every statistic in the same direction, and inverse dynamics every one except clustering by video. Pixel prediction shifts them in the opposite direction, requiring 445 principal components. With reconstruction alone, past conditioning produces smaller changes in the same direction as the temporal objectives (316 against 345 principal components). The maps and spectra illustrate the smoother spatial structure under temporal prediction (Figures 21–23).

Table 12: Geometry of frozen latent spaces on 1,000 held-out frames from 100 videos per experiment. Columns give principal components for 95% variance, participation ratio, latent-to-pixel turning-angle ratio, within-to-across-video distance ratio, spectral slope and high-frequency share (defined in the text). Lower values indicate lower dimension, straighter trajectories, tighter video clustering and smoother maps, respectively. Mean and standard error over three training seeds unless one is stated. Samples differ between experiments. Latent prediction reduces dimension and high-frequency content in both experiments; inverse dynamics also smooths the latents but does not tighten clustering by video.

	components for 95%	participation ratio	turning-angle ratio	within / across video distance	spectral slope	high-frequency spectral share
matched reconstruction, three seeds						
reconstruction only	388	52.7	1.100	0.423	−0.837	0.073
	±0.9	±0.03	±0.0001	±0.0004	±0.0003	±0.00008
latent prediction	177	49.4	1.019	0.344	−0.907	0.061
	±8	±1.1	±0.007	±0.005	±0.010	±0.002
pixel prediction	445	64.4	1.110	0.443	−0.763	0.094
	±0.3	±0.2	±0.0004	±0.001	±0.0008	±0.0002
contrastive, random negatives	293	46.7	1.086	0.384	−0.933	0.052
	±6	±0.7	±0.0009	±0.002	±0.010	±0.002
matched budget, past-conditioned decoder, three seeds						
reconstruction only	316	47.7	1.090	0.402	−0.886	0.063
	±10	±0.7	±0.002	±0.004	±0.008	±0.002
latent prediction	110	36.0	0.879	0.323	−0.948	0.049
	±1	±0.5	±0.002	±0.004	±0.004	±0.001
inverse dynamics	239	44.9	1.005	0.464	−1.020	0.043
	±4	±2.0	±0.001	±0.002	±0.009	±0.002
matched budget, ordinary decoder (reconstruction only: three seeds; pretrained: one)						
reconstruction only	345	49.8	1.092	0.407	−0.864	0.067
	±12	±0.8	±0.002	±0.005	±0.009	±0.002
pretrained, no fine-tuning	383	52.5	1.099	0.421	−0.838	0.072

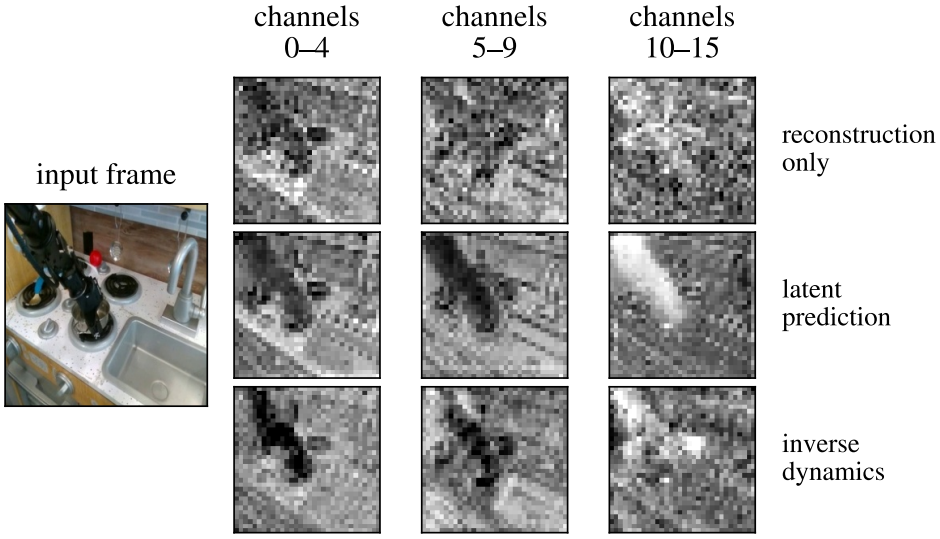


Figure 21: Latent maps of one held-out Bridge frame under the three past-conditioned encoders trained for 80,000 updates, training seed 1234. Columns average channels 0–4, 5–9 and 10–15 of each $16 \times 32 \times 32$ latent. Each map is standardized separately, with black low and white high; intensity is not comparable across maps. Latent prediction and inverse dynamics produce smoother regions around the arm and objects than reconstruction alone in this example.

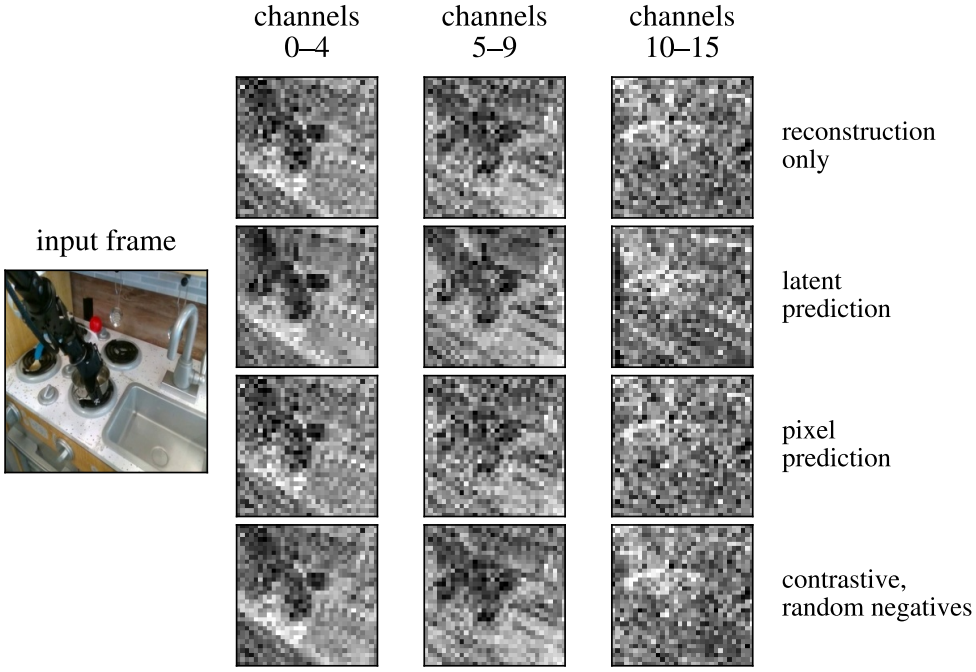


Figure 22: Latent maps of the same held-out Bridge frame under the four encoders trained to equal reconstruction quality, all with the ordinary decoder, training seed 0. Columns average channels 0–4, 5–9 and 10–15 of the $16 \times 32 \times 32$ latent; each map is standardized separately, with black low and white high. Latent and contrastive prediction suppress fine spatial variation compared with reconstruction alone, whereas pixel prediction retains it in this example.

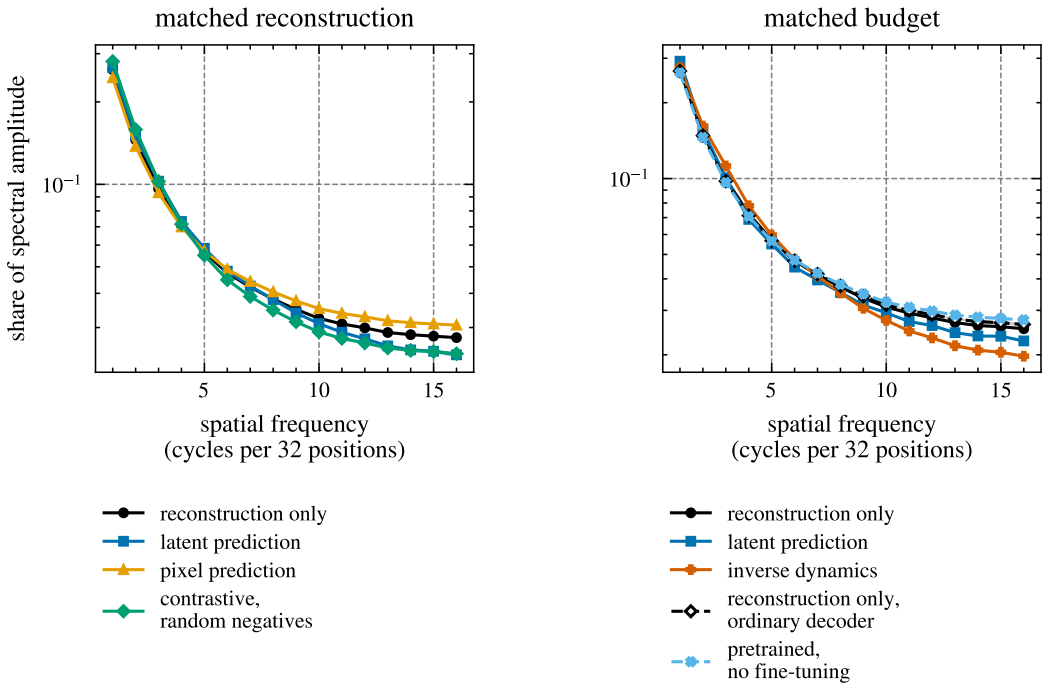


Figure 23: Spatial spectra of the frozen latents, normalized by total radial amplitude over frequencies 1–16 cycles per 32 positions. Left: matched reconstruction. Right: matched budget; dashed curves use the ordinary decoder. Curves and bands show the mean and standard error over three training seeds, except the pretrained encoder (one). Latent and contrastive prediction reduce high-frequency components at matched reconstruction; latent prediction and inverse dynamics do so at matched budget.

H.9 Ablations

The ablations use training seed 1234 unless stated otherwise.

H.9.1 Temporal weight

Figure 24 gives the world models on the past-conditioned autoencoders of Section 5.2 at $\lambda = 0.1$ to 0.5; at $\lambda = 0$ and 0.7 it uses those of Section 5.3. Lower weights trade part of the long-rollout gain of $\lambda = 0.7$ for distributional quality (Section 5.3): at $\lambda = 0.1$ to 0.5, latent prediction gives FID 12.8 to 14.3 and FVD 31.8 to 33.4, against 18.5 and 36.5 for reconstruction alone with the same decoder and seed.

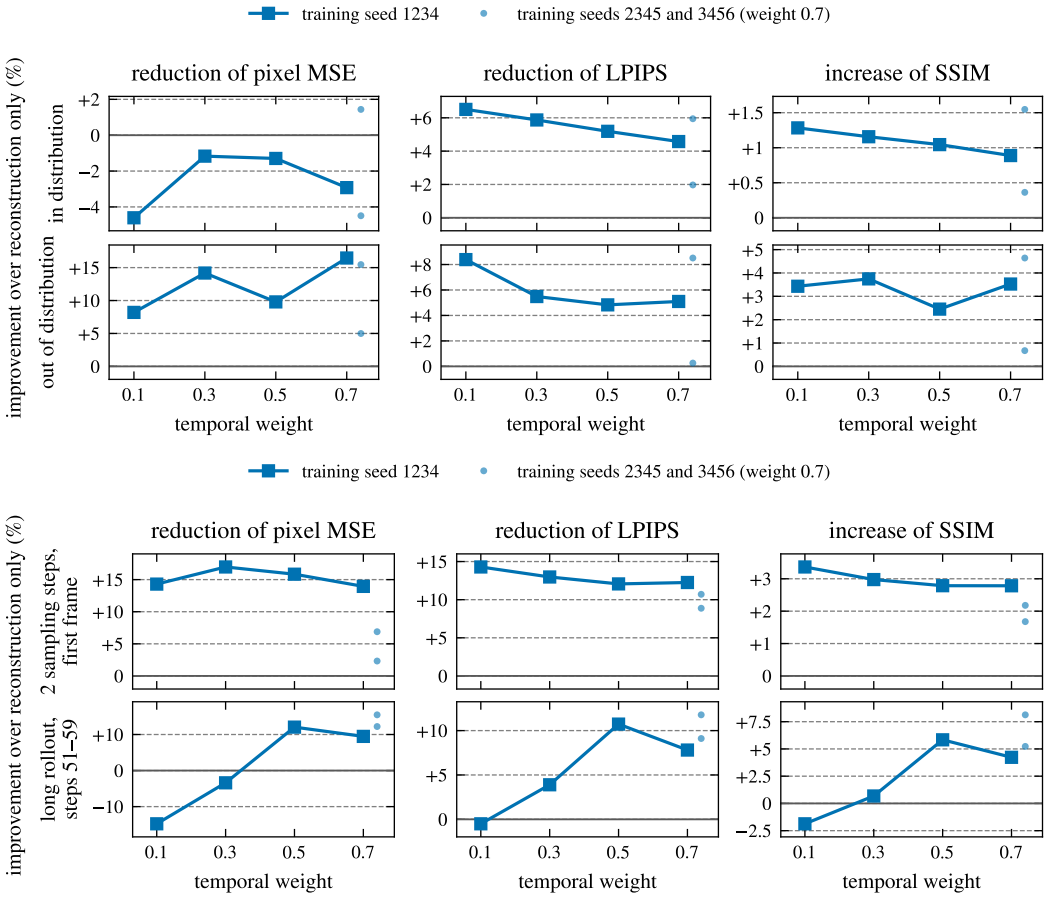


Figure 24: World-model improvement over reconstruction alone as the temporal weight varies, with the past-conditioned decoder fixed. Positive values are better: relative reductions of pixel MSE and LPIPS, and increases of SSIM. Rows show in-distribution, out-of-distribution, two-step first-frame and long-rollout evaluations. Lines: seed 1234; dots: seeds 2345 and 3456 at weight 0.7. Weights 0.1–0.5 use one seed. Panels have independent scales. LPIPS improves at every weight in the short-rollout and few-step evaluations; long-rollout gains favor larger weights.

H.9.2 A semantic encoder

We replace the encoder of FLUX by DINOv2-Base (Oquab et al., 2024) with registers (Darcet et al., 2024) and use the pretrained ViT-XL decoder of RAE (Zheng et al., 2026) (the `wReg_base/ViTXL_n08` checkpoint). The encoder resizes each frame to 224×224 with bicubic interpolation, applies ImageNet normalization and produces a $768 \times 16 \times 16$ latent; the decoder reconstructs at 256×256. We retain RAE's released latent-normalization statistics and add no latent

noise during fine-tuning. The decoder has 28 layers of width 1,152, 16 attention heads and feed-forward width 4,096. We add seven zero-output-initialized cross-attention adapters, with eight heads, after layers 1, 5, 9, 13, 17, 21 and 25. Their 256 context tokens, of width 768, fuse projected image patches with projected features from a frozen copy of the pretrained encoder, as in the FLUX decoder. The encoder, decoder and adapters are fine-tuned from the first update; the latent predictor has the architecture of Appendix C, with input and output channels changed to 768. We use latent-prediction weights 0, 0.1, 0.3 and 0.7, training seed 1234, and the matched-budget autoencoder optimization (batch 48, learning rate 3.46×10^{-6}, 80,000 updates).

The world model uses one token per latent position (patch size 1), 12 backbone layers of width 768 with 12 heads, and two output layers of width 2,048 with 16 heads. It predicts the clean latent; its output is converted to velocity for the training loss. It standardizes every channel and spatial position separately, using statistics computed on 100 training batches. Both the training noise indices and the DDIM grid are warped by $s \mapsto 7s/(1+6s)$ for normalized index $s \in [0,1]$, then rounded to one of the 1,000 noise levels. The world model uses the matched-budget optimization (effective batch 16, learning rate 7×10^{-5}, 3,000-update warm-up and 100,000 updates); the evaluation sets, guidance scale and numbers of sampling steps are those of Appendix E. At weight 0.1, latent prediction improves every metric. At weights 0.3 and 0.7, it still lowers the out-of-distribution pixel MSE and the latent forecast error, but it worsens the in-distribution errors, FID (by 52% and 162%) and FVD (by 40% and 116%). With a semantic encoder, a small weight is therefore preferable.

Table 13: Temporal prediction with a DINOv2 encoder and past-conditioned RAE decoder: 80,000 autoencoder and 100,000 world-model updates, training seed 1234. The reference uses reconstruction alone; other rows give percentage changes against it. Reference values are absolute and pixel MSE is unscaled. Green/orange shading marks improvement/worsening; bold marks the best value in each column. FID, FVD and IS measure distributional quality. Weight 0.1 improves every metric, whereas larger weights further lower latent forecast error but worsen several video-quality metrics.

	In distribution			Out of distribution		
	pixel MSE ↓	LPIPS ↓	SSIM ↑	pixel MSE ↓	LPIPS ↓	SSIM ↑
reconstruction only (reference)	0.01078	0.158	0.797	0.03511	0.278	0.698
latent future prediction, weight 0.1	**−2.0%**	**−5.2%**	**+1.0%**	−13.7%	**−7.5%**	**+2.5%**
latent future prediction, weight 0.3	+8.2%	+2.7%	−0.9%	−15.4%	−2.6%	+0.8%
latent future prediction, weight 0.7	+11.1%	+12.3%	−4.3%	**−19.7%**	+7.6%	−3.6%

	2 sampling steps			Long rollout, steps 51–59		
	pixel MSE ↓	LPIPS ↓	SSIM ↑	pixel MSE ↓	LPIPS ↓	SSIM ↑
reconstruction only (reference)	0.01101	0.160	0.800	0.03094	0.289	0.687
latent future prediction, weight 0.1	**−1.7%**	**−6.1%**	**+1.0%**	**−13.6%**	**−8.2%**	**+4.2%**
latent future prediction, weight 0.3	+10.7%	+3.0%	−1.0%	+4.7%	+1.0%	−0.2%
latent future prediction, weight 0.7	+19.0%	+15.0%	−4.9%	+7.1%	+2.9%	−2.9%

	Latent forecast error ↓	
	in distribution	out of distribution
reconstruction only (reference)	0.522	0.531
latent future prediction, weight 0.1	−70.2%	−68.4%
latent future prediction, weight 0.3	−88.9%	−87.7%
latent future prediction, weight 0.7	**−95.1%**	**−93.8%**

	Distribution metrics		
	FID ↓	FVD ↓	IS ↑
reconstruction only (reference)	8.7	26.0	4.91
latent future prediction, weight 0.1	**−13.7%**	**−17.0%**	+1.4%
latent future prediction, weight 0.3	+52.2%	+40.1%	**+1.9%**
latent future prediction, weight 0.7	+162.2%	+116.4%	−1.3%

H.9.3 WORLD-MODEL SIZE

On a second pair of autoencoders trained with the matched-budget recipe (reconstruction only, and latent prediction at weight 0.7; one training run each), we train world models of width 384 with 12 layers and of width 1,024 with 16 layers (Figure 25). Out of distribution and with 2 sampling steps, latent prediction improves the world model at both widths: out of distribution, it lowers the pixel MSE by 27.7% at width 384, and by 24.5% and 16.8% at width 1,024 after 99,540 and 200,000 updates. In distribution, its pixel MSE is 2 to 5% higher at both widths, and its LPIPS is lower at width 384 and 1 to 2% higher at width 1,024; on long rollouts at width 1,024, it lowers the pixel MSE by 5.1% and 2.4% but barely changes LPIPS.

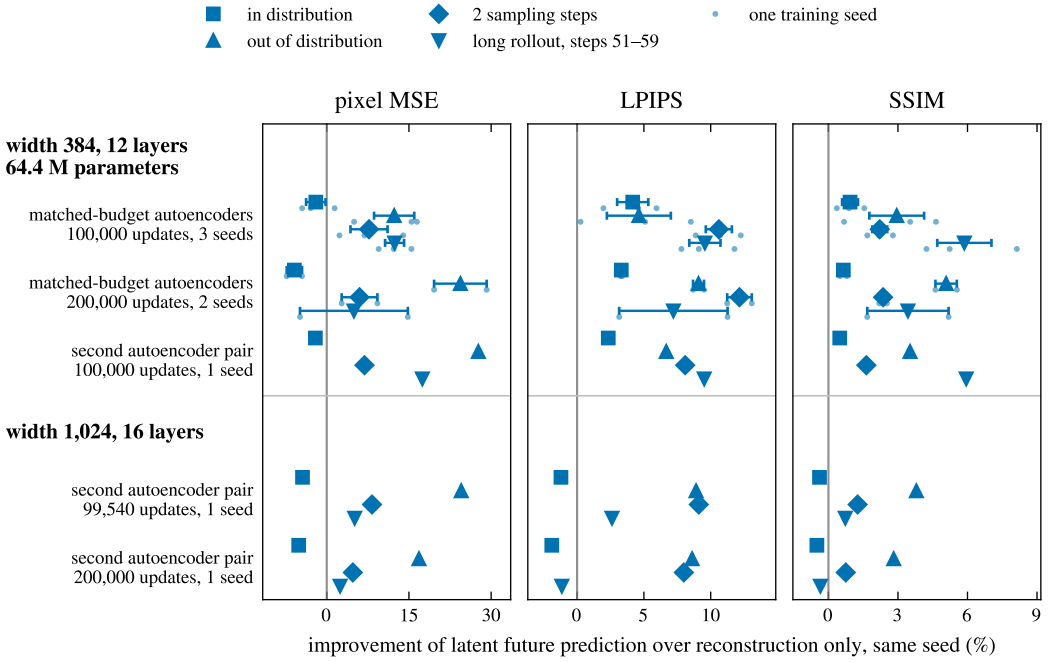


Figure 25: Effect of latent future prediction across world-model sizes, against reconstruction alone with the past-conditioned decoder and the same seed. Positive values are better: relative reductions of pixel MSE and LPIPS, and increases of SSIM. Rows specify width, training updates and autoencoder pair; markers identify the four evaluation conditions. Matched-budget pair: mean and standard error over training seeds, with small dots for individual seeds. Second pair: one run per row. Gains persist out of distribution and with two sampling steps at both widths, while in-distribution and long-rollout effects are mixed.

H.9.4 KL PENALTY

Our autoencoders are fine-tuned without the KL penalty of latent diffusion (Appendix B). We add it with weight 10^{-6} to the matched-budget recipe, with and without latent prediction (Table 14). With the penalty, latent prediction still improves on reconstruction only on every error metric, most out of distribution (pixel MSE -30.5% against reconstruction only with the penalty), while its FVD is 8.2% worse. The penalty itself improves reconstruction only on most metrics (Table 14).

Table 14: KL regularization under the matched-budget recipe, with the past-conditioned decoder and latent prediction at weight 0.7. Autoencoders are trained with or without a KL penalty of 10^{-6}; all runs use training seed 1234. Reference rows give reconstruction alone without KL, in absolute units with unscaled pixel MSE; other rows give percentage changes against that reference. Green/orange shading marks improvement/worsening; bold marks the best value; –: not measured. With KL, latent prediction still improves all framewise metrics over reconstruction alone with KL, while its FVD is worse.

	In distribution			Out of distribution		
	pixel MSE ↓	LPIPS ↓	SSIM ↑	pixel MSE ↓	LPIPS ↓	SSIM ↑
no KL penalty						
reconstruction only (reference)	0.01376	0.178	0.801	0.03181	0.274	0.713
latent future prediction	+2.9%	−4.6%	+0.9%	−16.5%	−5.1%	+3.5%
KL penalty, weight 10^{-6}						
reconstruction only	−0.4%	−5.3%	+1.1%	+14.3%	−5.9%	+1.8%
latent future prediction	**−1.5%**	**−7.9%**	**+1.6%**	**−20.6%**	**−12.1%**	**+5.4%**

	2 sampling steps			Long rollout, steps 51–59		
	pixel MSE ↓	LPIPS ↓	SSIM ↑	pixel MSE ↓	LPIPS ↓	SSIM ↑
no KL penalty						
reconstruction only (reference)	0.01787	0.203	0.782	0.04527	0.373	0.620
latent future prediction	−13.9%	−12.3%	+2.8%	**−9.5%**	**−7.8%**	**+4.2%**
KL penalty, weight 10^{-6}						
reconstruction only	−6.7%	−7.8%	+1.8%	−3.8%	−2.5%	+1.1%
latent future prediction	**−20.4%**	**−16.5%**	**+3.8%**	−7.6%	−6.8%	+3.2%

	Latent forecast error ↓	
	in distribution	out of distribution
no KL penalty		
reconstruction only (reference)	0.227	0.193
latent future prediction	**−79.4%**	**−72.7%**
KL penalty, weight 10^{-6}		
reconstruction only	–	–
latent future prediction	–	–

	Distribution metrics		
	FID ↓	FVD ↓	IS ↑
no KL penalty			
reconstruction only (reference)	18.5	36.5	**5.01**
latent future prediction	+11.7%	+20.0%	−2.3%
KL penalty, weight 10^{-6}			
reconstruction only	−16.8%	**−3.2%**	−0.7%
latent future prediction	**−20.2%**	+4.7%	−1.7%