SPT-FLOW: ONE INTEGRATION, MANY FUTURES IN CONTINUOUS-TIME WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Continuous-time world models generate a latent trajectory through a single ODE integration, but adding stochastic conditioning does not ensure that the model uses it. In our teacher-forced baseline, posterior means vary by only about 1% of the prior scale, and matched and swapped latents yield indistinguishable reconstruction PSNR at the reported precision. We study a supervision-shortcut hypothesis: ground-truth intermediate states can reduce the need for trajectory-associated latent information. We introduce SPT-FLOW, which samples one latent per trajectory and matches a free-running ODE rollout to demonstration states at multiple readout times. Re-anchoring experiments support the proposed mechanism. On LIBERO, with both models trained for 30k optimizer steps, SPT-FLOW improves best-of-10 PSNR over our retrained ODEWorld baseline by 2.0 and 2.4 dB at the mid-trajectory and final frames and reduces mean LPIPS by 23%, with comparable per-sample inference cost. The rollout control recovers most of the accuracy gain; learned latent conditioning adds diversity and best-of-K gains while reducing mean sample accuracy relative to this control. The diagnostics support trajectory-associated latent utilization; semantic mode coverage under matched conditions in the robotic setting remains unverified.

1 INTRODUCTION

World models support robotic manipulation by predicting future observations conditioned on an initial observation and a goal (Du et al., 2023; Black et al., 2024; Hu et al., 2025). Many approaches use discrete stochastic (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2025) or deterministic (Hansen et al., 2024; Zhou et al., 2025) latent transitions. ODEWorld (Liu et al., 2026) instead models latent dynamics as an ordinary differential equation in physical time, generating a trajectory through a single numerical integration with arbitrary-time readout.

This formulation nevertheless produces only one trajectory for a fixed initial state and goal. Manipulation tasks can admit multiple valid continuations: *put both pots on the stove*, for example, permits either pot to be moved first. When the velocity-field inputs cannot distinguish these alternatives, squared-error supervision favors their conditional mean, whose integration may match neither continuation. Appendix E illustrates this failure in a controlled setting and presents distinct continuations recorded on a physical robot arm.

Conditional latent variables offer a standard approach to multimodal prediction (Sohn et al., 2015), but improved accuracy does not guarantee informative latent representations. In our experiments, adding a trajectory-level latent ϵ improves best-of-K accuracy while posterior means vary little across trajectories and swapping inferred latents barely affects reconstruction accuracy (Section 3.4). We term this behavior *decorative latent conditioning*.

A potential explanation is a supervision shortcut: ground-truth intermediate states can reveal the demonstrated continuation, allowing the velocity field to predict its target without using ϵ. At inference, these states are unavailable, yet training has not explicitly required the latent to resolve the remaining ambiguity.

We address this mismatch with SPT-FLOW, which replaces teacher-forced velocity supervision with rollout matching. During training, a latent sampled from the approximate posterior conditions a free rollout from z_0, whose predicted states are matched to the demonstration. Removing intermedi-

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.

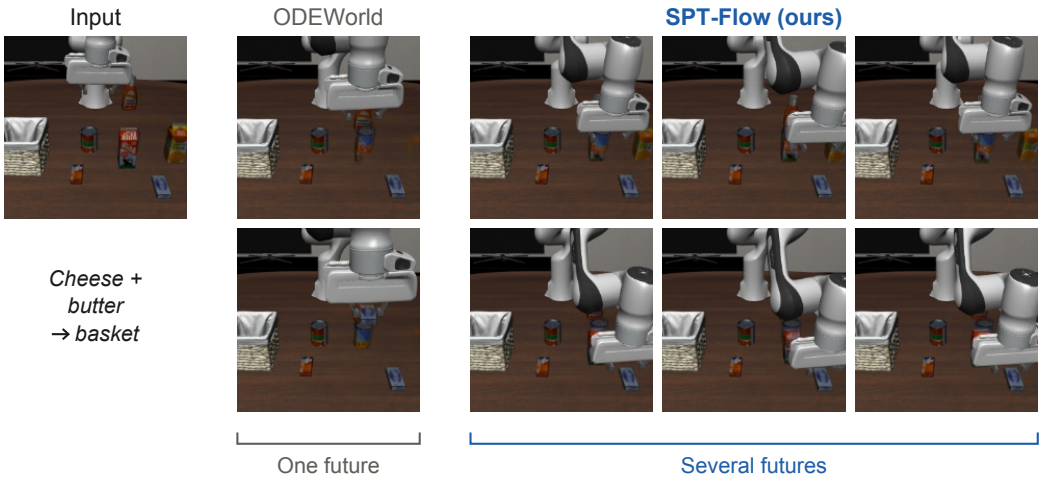

Figure 1: Predictions conditioned on the same initial frame and instruction. ODEWorld produces one trajectory; SPT-FLOW samples three, each through one integration. Rows show $t/T = 0.6$ and 1.0. The example is selected for diversity; model settings follow Figure 5.

ate ground-truth inputs encourages the model to encode and use trajectory-associated information through ϵ. At inference, different prior samples generate different trajectories, each through a single integration (Figure 1). Here, multiple futures denotes sampled trajectories; the LIBERO diagnostics assess latent utilization, while recovery of known continuation modes is tested in the synthetic setting of Appendix E.

Contributions.

- We introduce SPT-FLOW, combining trajectory-level latent conditioning with rollout matching for stochastic physical-time prediction. It preserves the deterministic baseline's number of velocity-field evaluations per trajectory and achieves comparable inference latency.
- We characterize decorative latent conditioning using posterior specificity and latent swapping. Under rollout matching, specificity exceeds the predefined threshold in all three seeds, and matched latents outperform swapped latents by 1.05 dB. Controlled re-anchoring experiments provide evidence for the proposed supervision shortcut.
- On LIBERO, at 30k optimizer steps, SPT-FLOW improves best-of-10 PSNR over our retrained ODEWorld baseline by 2.0 and 2.4 dB, and mean sample PSNR by 1.0 and 1.3 dB, at the mid-trajectory and final frames, respectively, while reducing mean LPIPS by 23%. The rollout control recovers most of the accuracy gain; learned latent conditioning improves best-of-K performance and diversity at the expense of mean sample accuracy relative to that control. The prediction advantage persists when both methods receive four times the standard training steps.

2 RELATED WORK

World models. World models support manipulation through goal-conditioned visual prediction (Du et al., 2023; Black et al., 2024; Hu et al., 2025; Huang et al., 2025b). Many approaches use discrete stochastic (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2025) or deterministic (Hansen et al., 2024; Zhou et al., 2025) latent transitions. Continuous-time alternatives include neural ODEs (Chen et al., 2018), latent ODEs (Rubanova et al., 2019), neural SDEs (Li et al., 2020), and continuous-

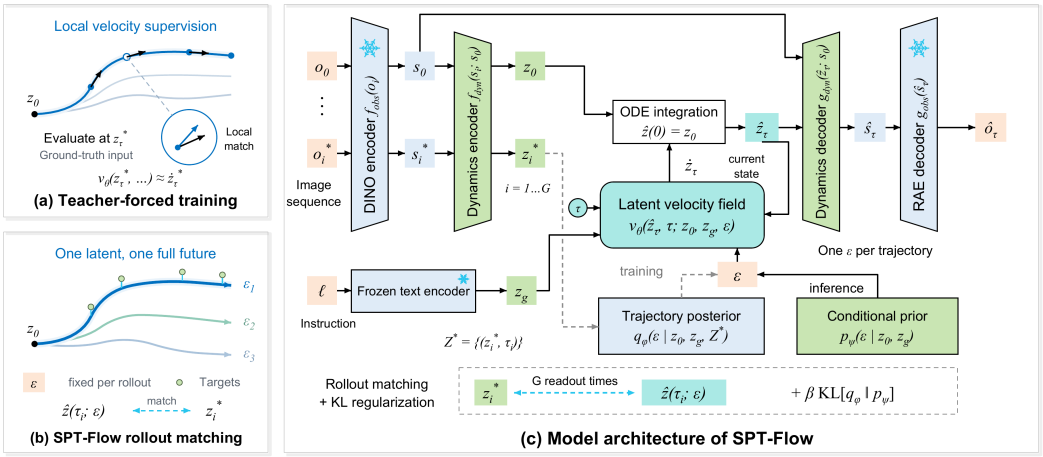

Figure 2: SPT-FLOW and its training objective. **(a)** Teacher-forced velocity supervision (Equation 2) evaluates the field at a ground-truth latent $z_\tau^\star$, which can reveal the demonstrated continuation. **(b)** Rollout matching (Equation 4) samples one ϵ, integrates from z_0, and supervises predictions at G readout times without intermediate ground-truth inputs. **(c)** Architecture: snowflakes indicate frozen modules and dashed paths are used only during training. At inference, ϵ is sampled from $p_\psi(\epsilon \mid z_0, z_g)$.

time video models (Park et al., 2021; Ye & Bilodeau, 2024). ODEWorld (Liu et al., 2026) generates latent trajectories by integration in physical time.[1] Recent stochastic world models include object-centric latent dynamics (Daniel et al., 2026) and flow matching in pretrained feature spaces (Porcher et al., 2026). SPT-FLOW builds on ODEWorld's representation and single-integration formulation, introducing trajectory-level latent conditioning trained through rollout matching.

Latent conditioning and utilization. Under standard uniqueness conditions, deterministic ODE dynamics produce one trajectory for a fixed initial state and conditioning (Dupont et al., 2019). Latent conditioning provides one way to represent multiple futures, as explored in neural ODEs (Norcliffe et al., 2021; Cheng et al., 2024) and stochastic video prediction (Babaeizadeh et al., 2018; Lee et al., 2018; Denton & Fergus, 2018; Franceschi et al., 2020; Ye et al., 2019). The same argument underlies generative action heads, where a sample from a diffusion or flow-matching policy selects an action chunk instead of regressing to the mean of the demonstrated ones (Chi et al., 2023; Black et al., 2025; Hu et al., 2024; Zhai et al., 2025), and Guo & Schwing (2025) attach a latent to a flow-matching field for the same reason in generative flow time. However, variational latents may remain uninformative because of optimization dynamics, decoder conditioning, or nonidentifiability (Bowman et al., 2016; Chen et al., 2017; Wang et al., 2021; Dang et al., 2024); a further reading attributes the neglect to a decoder that sees the ground truth throughout training (De Fauw et al., 2019), or to data that pair each input with a single recorded future (Zhou et al., 2021). Proposed mitigations include KL scheduling, free bits, auxiliary objectives, and restrictions on decoder inputs (Sønderby et al., 2016; Kingma et al., 2016; Goyal et al., 2017; Roberts et al., 2018; Miladinović et al., 2022). Closely related, Sip et al. (2025) examine how teacher-forcing intervals affect the use of stochastic inputs in dynamical VAEs. We study this interaction in physical-time world models, distinguishing sampling-related accuracy gains from trajectory-associated latent utilization through posterior specificity and latent swapping.

Rollout supervision. The mismatch between training on observed states and predicting from generated states is known as exposure bias (Ranzato et al., 2016) and motivates scheduled sampling (Bengio et al., 2015; Huszár, 2015), data aggregation (Venkatraman et al., 2015; Talvitie, 2017), adversarial matching (Lamb et al., 2016), prediction-dependent transitions in recurrent world models (Chiappa et al., 2017), and training with self-generated context (Chen et al., 2024; Huang et al.,

[1]We distinguish posterior collapse of a variational latent from the representation collapse discussed in ODEWorld.

2025a). Latent overshooting extends variational training to multi-step predictions (Hafner et al., 2019), while RDR supervises decoded free-running trajectories in a deterministic latent world model (Shah & Shrestha, 2026). Our focus is the effect of rollout supervision on a trajectory-level variational latent, beyond improvements in prediction accuracy. Re-anchoring experiments vary access to ground-truth states while holding the rollout posterior fixed. Additional implementation differences between teacher-forced and rollout training are documented in Appendix A.

3 SPT-FLOW: A STOCHASTIC PHYSICAL-TIME FLOW

3.1 PHYSICAL-TIME FLOW UNDER TEACHER-FORCED SUPERVISION

Given an initial observation o_0 and instruction ℓ, a frozen image encoder produces tokens $s_t = \mathrm{enc}(o_t)$, and a decoupling module maps (s_0, s_t) to $z_t = \mathrm{dec}(s_0, s_t)$. A frozen instruction encoder provides z_g, and a frozen decoder reconstructs frames. ODEWorld's physical-time flow (PT-Flow) evolves these latents through

$$\frac{\mathrm{d}z}{\mathrm{d}\tau} = v_\theta(z_0, z, z_g, \tau), \qquad z(0) = z_0, \tag{1}$$

A single numerical integration generates the trajectory, with readouts at arbitrary query times. We use fixed-step Runge–Kutta integration at inference (Appendix A). Teacher-forced training evaluates the field at demonstration states $z^\star_\tau$ and matches a target velocity $\dot{z}^\star_\tau$:

$$\mathcal{L}_{\mathrm{tf}} \;=\; \big\| v_\theta\big(z_0, \underbrace{z^\star_\tau}_{\text{ground truth}}, z_g, \tau\big) - \dot{z}^\star_\tau \big\|^2. \tag{2}$$

Appendix A details target construction and differences from the released model. Crucially, the field receives ground-truth states during training but its own predictions during inference.

For fixed initial conditions and goal, the deterministic ODE produces one trajectory. Where its inputs do not distinguish alternative continuations, squared-error supervision favors their conditional mean velocity (Bishop, 1994; Mathieu et al., 2016), which can yield a rollout inconsistent with either alternative.

3.2 THE TRAJECTORY-LEVEL LATENT AND THE SUPERVISION SHORTCUT

To represent multiple futures, we condition the field on ϵ, held fixed throughout each rollout: $v_\theta(z_0, z, z_g, \epsilon, \tau)$. Under teacher forcing, the posterior $q(\epsilon \mid z_0, z^\star_\tau, z_g, \tau)$ infers the latent from a demonstration state; a learned prior $p(\epsilon \mid z_0, z_g)$ supplies samples at inference. A KL penalty regularizes the posterior toward the prior.

However, the field is evaluated at $z^\star_\tau$ (Figure 2a), and that state can already identify the demonstrated continuation, allowing the field to predict its velocity without informative latent conditioning. This possible shortcut reduces the incentive to encode continuation-specific information in ϵ. It does not guarantee latent neglect, as our low-dimensional example shows (Appendix E). In our main experiments, however, posterior means vary little across trajectories, and swapping inferred latents provides no measurable accuracy change (Section 4.3). We call this *decorative latent conditioning*: sampling improves best-of-K accuracy despite limited evidence of trajectory-specific encoding and use.

3.3 ROLLOUT MATCHING

SPT-FLOW instead supervises freely integrated trajectories. For readout times $\tau_1 < \cdots < \tau_G$ and demonstration states $z^\star_{1:G}$, a permutation-invariant posterior pools the state–time pairs and produces one sample:

$$q\big(\epsilon \,\big|\, z_0, z_g, \{(z^\star_i, \tau_i)\}_{i=1}^G\big), \tag{3}$$

Conditioned on this sample, the field integrates from z_0 without receiving intermediate ground-truth states. We match its predictions $\hat{z}(\tau_i; \epsilon)$ to the demonstration:

$$\mathcal{L}_{\mathrm{roll}} \;=\; \frac{1}{G}\sum_{i=1}^{G} \big\| \hat{z}(\tau_i; \epsilon) - z^\star_i \big\|^2 \;+\; \beta\, \mathrm{KL}[q \,\|\, p]\,, \tag{4}$$

We use $G = 4$ readout times per trajectory, and the KL regularizer uses a per-dimension free-bits floor. Solver settings, hyperparameters, and differences in posterior inputs and KL aggregation between training variants are documented in Appendix A.

Conditional on z_0 and z_g, ϵ is the only input through which the demonstrated future enters the rollout (Figure 2b). Removing intermediate ground-truth conditioning therefore encourages the posterior to encode information useful for reconstructing that future. At inference, we sample ϵ from the prior, integrate once, and decode. Training and inference share the frozen modules and velocity field in Figure 2c; the posterior supplies the training latent and the prior supplies the inference latent. Solver settings differ as detailed in Appendix A. Each sampled trajectory uses the same number of velocity-field evaluations as PT-Flow, with comparable measured latency (Section 4.2); generating K samples requires K integrations.

3.4 Posterior specificity and the latent swap

Because best-of-K gains alone do not establish informative latent conditioning, we apply two complementary diagnostics to trained models. Inspired by the active-units statistic of Burda et al. (2016), posterior specificity measures the across-trajectory standard deviation of each posterior mean, normalized by the corresponding average prior standard deviation and averaged over dimensions. This normalization accounts for our learned conditional prior. Low specificity indicates limited variation in posterior means, but does not exclude information in other posterior statistics.

Inspired by latent control swapping (Seitzer et al., 2024), the latent swap compares rollouts conditioned on latents inferred from the target trajectory and an unrelated trajectory, holding the initial state and goal fixed. Both are scored against the same demonstration. An advantage for the matched latent supports the use of trajectory-associated information; equal scores indicate no measurable benefit under this intervention, rather than independence from ϵ. Such information may include task, scene, or timing differences; this intervention does not isolate continuation choice under matched initial conditions and instructions. Both diagnostics require one evaluation pass over held-out trajectories and no retraining. Section 4.3 reports thresholds, sample sizes, and results.

4 Experiments

4.1 Setup

We evaluate on LIBERO (Liu et al., 2023), using $5{,}000$ trajectories across 100 tasks, re-rendered at 256×256 by replaying stored simulator states. We allocate $4{,}500$ trajectories to training and 250 each to fixed development and test sets; the test set is excluded from model development. Unless otherwise stated, models train for 30k steps, evaluation uses the full test set with $K = 10$ conditional-prior samples per stochastic model, and results report the mean and standard deviation across three seeds. Seed spreads are normalized by n rather than $n - 1$, since they describe the dispersion of three runs rather than estimate a population, and the one development-split result is labeled where it appears. We retrain the released ODEWorld architecture using Equation 2, since the release provides inference code only; all models share the same frozen encoders and decoder. Training comparisons match optimizer steps; training NFE and GPU-hours are not reported, so these comparisons do not establish equal-compute performance.

PSNR is evaluated against recorded observations at mid-trajectory and final frames, reporting both best-of-K and mean sample performance. Best-of-K follows the selection principle of variety loss (Gupta et al., 2018) and may favor dispersed predictions (Thiede & Brahma, 2019), motivating complementary mean reporting. These frame-level metrics do not establish whole-trajectory consistency. Diversity is the mean pairwise distance between terminal latent states and measures representation-space variation. All frames are compared after decoding with the shared frozen decoder, so differences reflect the predicted latent trajectory rather than the decoder. LPIPS and FVD are reported in Section 4.4; implementation details and baseline deviations are documented in Appendix A.

Table 1: Results on the LIBERO test set (250 trajectories), with $K = 10$ samples. PSNR is reported in dB; diversity denotes mean pairwise terminal latent distance. Both training budgets use three seeds; standard deviations are shown for 30k steps, and means only for 60k steps. The rollout control is trained with $\epsilon = 0$ but retains random conditioning at evaluation, as discussed in the text. Bold indicates the highest PSNR within each training budget.

	mid-trajectory		final frame		
Model	best-of-10	mean	best-of-10	mean	diversity
ODEWorld (PT-Flow)	17.81±.22	17.81±.22	15.98±.17	15.98±.17	0.00
+ latent, teacher-forced	18.22±.15	17.86±.23	16.20±.24	15.87±.26	1.46
+ rollout control	19.33±.06	**19.02**±.05	18.18±.20	**17.77**±.19	0.51
SPT-FLOW (both)	**19.82**±.10	18.85±.10	**18.42**±.08	17.29±.10	**2.03**
60k steps, three seeds per row					
ODEWorld (PT-Flow)	18.67	18.67	16.35	16.35	0.00
+ latent, teacher-forced	18.74	18.45	16.69	16.35	1.22
+ rollout control	19.61	**19.16**	18.64	**18.07**	0.51
SPT-FLOW (both)	**19.80**	19.07	**18.74**	17.78	1.18

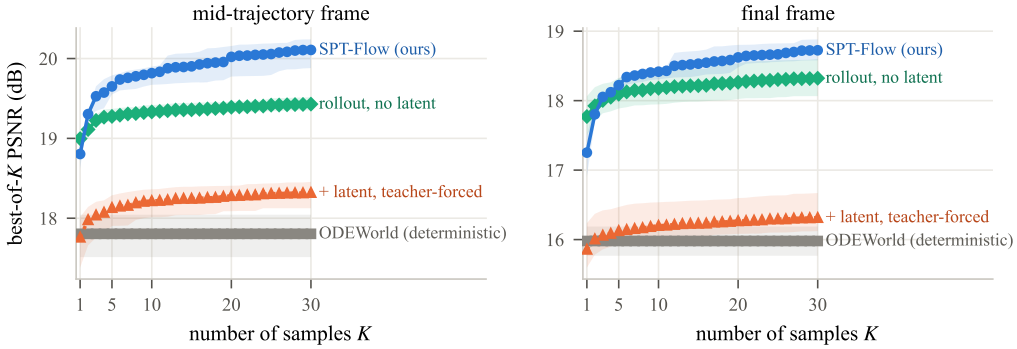

Figure 3: Best-of-K PSNR as a function of sample count at 30k training steps. Lines show means across three seeds, and shaded bands indicate the minimum-to-maximum seed range. Rollout matching improves performance at $K = 1$, while SPT-FLOW exhibits larger gains as K increases. ODEWorld is independent of K.

4.2 ACCURACY AND DIVERSITY AT COMPARABLE INFERENCE COST

At 30k steps, SPT-FLOW improves best-of-10 PSNR over ODEWorld by 2.01 and 2.44 dB at the mid-trajectory and final frames, respectively (Table 1). Rollout matching without learned latent conditioning recovers 1.52 and 2.20 dB, while SPT-FLOW exceeds this control by 0.49 and 0.24 dB. Mean sample PSNR also exceeds ODEWorld by 1.04 and 1.31 dB, although it remains 0.17 and 0.48 dB below the rollout control. At 60k steps, the best-of-10 advantage over ODEWorld persists at 1.13 and 2.39 dB. Learned latent conditioning increases terminal diversity to 2.03, compared with 1.46 under teacher forcing and 0.51 for the rollout control; ODEWorld has none, since a deterministic field integrates to one trajectory per initial state and goal. The control's residual diversity arises from evaluation with an untrained prior despite training at $\epsilon = 0$; it is therefore not strictly deterministic. The reported increments are relative to this control; they do not isolate the latent's contribution against rollout evaluation at $\epsilon = 0$.

The sampling curves further distinguish the two contributions (Figure 3): rollout matching improves accuracy at $K = 1$, by 1.2 and 1.8 dB at the two frames, while increasing K from 1 to 30 yields gains of 1.30 and 1.47 dB for SPT-FLOW, versus 0.43 and 0.55 dB for the control. At the final frame, SPT-FLOW overtakes the control at $K = 3$ and achieves a 0.41 dB advantage at $K = 30$.

Per-sample inference costs remain comparable: a 50-frame latent rollout takes 151.1 ms for SPT-FLOW versus 149.9 ms for ODEWorld on the same GPU, an approximately 0.8% increase, with 200 velocity-field evaluations for both. The latent is sampled once and held fixed throughout integration. Best-of-K evaluation nevertheless requires K rollouts, with computational cost scaling approximately linearly under sequential execution.

Table 2: Both diagnostics of Section 3.4 are applied to trained models on held-out trajectories. Specificity measures posterior-mean variation normalized by the prior scale; the threshold 0.25 was fixed before evaluation. *Own* and *swapped* use latents inferred from the target trajectory and another trajectory, respectively. PSNR is measured at the true temporal midpoint and is comparable within this evaluation protocol. At 30k steps, PSNR uses 250 trajectories across three seeds; specificity ranges use 40 trajectories. The 60k and low-floor settings use seed 0 and 40 trajectories.

Training	specificity	verdict	prior best-of-10	own	swapped
teacher-forced, 30k	0.007–0.012	not specific $(0/3)$	17.71	17.34	17.34
teacher-forced, 60k	0.010	not specific	18.37	18.04	18.04
teacher-forced, low KL floor	0.002	not specific	17.56	17.10	17.10
rollout matching, 30k	$\mathbf{0.306}$–$\mathbf{0.376}$	specific $(3/3)$	19.45	$\mathbf{18.96}$	17.91

Table 3: Perceptual and distributional evaluation across three seeds (mean $\pm$ standard deviation; lower is better). LPIPS (Zhang et al., 2018) is evaluated at the mid-trajectory frame on 50 held-out trajectories with $K = 10$. FVD (Unterthiner et al., 2018) uses 200 recorded reference clips and 50 generated clips for ODEWorld or 100 for each other model. The rollout control has no learned trajectory latent but retains random conditioning during evaluation.

	ODEWorld	rollout control	SPT-FLOW
LPIPS, mean of K	$0.181\pm.008$	$\mathbf{0.129}\pm.001$	$0.139\pm.003$
LPIPS, best of K	$0.181\pm.008$	$0.122\pm.001$	$\mathbf{0.116}\pm.004$
FVD	265 ± 41	210 ± 5	$\mathbf{187}\pm9$

4.3 LATENT SPECIFICITY AND UTILIZATION

Teacher-forced training yields posterior specificity of 0.007–0.012, corresponding to posterior-mean variation of approximately 1% of the prior scale (Table 2). Latents inferred from the target and unrelated trajectories produce indistinguishable PSNR at the reported precision across all three seeds. Neither doubling the training budget nor lowering the KL floor changes this pattern in the tested configurations. In contrast, rollout matching exceeds the predefined specificity threshold of 0.25 in every seed, with all 32 dimensions above 0.29, whereas no dimension of the teacher-forced posterior exceeds 0.022 in any seed (Appendix G). Using the target trajectory's inferred latent improves PSNR by 1.05 dB over the swapped latent, with a per-seed range of 0.78 to 1.19 dB (Table 4), providing evidence that trajectory-associated information contributes to prediction.

These results support the distinction between decorative and informative latent conditioning introduced in Section 3.2: best-of-K gains can coexist with weak posterior specificity and no measurable advantage from the trajectory-matched latent. However, the diagnostics do not establish that the latent selects among distinct continuations within a task. The association between task-level branching and the latent's accuracy gain is weak, with Spearman correlations of 0.005 and 0.115 at the mid-trajectory and final frames (Appendix G). We therefore interpret the results as evidence of trajectory-associated latent utilization, without attributing that information specifically to within-task continuation selection.

4.4 PERCEPTUAL AND DISTRIBUTIONAL QUALITY

Perceptual evaluation shows a pattern consistent with the PSNR results (Table 3). The rollout control reduces mean LPIPS from 0.181 to 0.129, a 29% improvement over ODEWorld. Adding learned latent conditioning further reduces best-of-K LPIPS from 0.122 to 0.116, a 36% reduction against ODEWorld, while increasing mean LPIPS by 0.010. With both rollout-based models evaluated using 100 generated clips, SPT-FLOW has a lower estimated FVD, 187 versus 210, approximately 11%. These results indicate that rollout matching improves average perceptual accuracy, while learned latent conditioning improves best-of-K performance. The FVD estimates are exploratory distributional measurements and do not establish semantic mode coverage. Comparisons with ODEWorld's FVD remain qualified by the unequal generated sample counts, 50 clips against 100 for each rollout-trained model, against the same 200 recorded references.

Table 4: Ground-truth states are withdrawn from the training rollout in three stages. Rollout variants share the same posterior; teacher forcing follows Equation 2. Specificity is reported as the mean [minimum, maximum] across three seeds. PSNR and diversity are seed means. The swap gap is own-minus-swapped mid-trajectory PSNR in dB, measured on seed 0 except for SPT-FLOW, which reports the three-seed mean and range.

Ground truth enters at	specificity	swap gap	mid best-of-10	final best-of-10	div.
every training signal (teacher forced)	0.009 [.007, .012]	0.00	18.22	16.20	1.46
every readout of the rollout	0.236 [.228, .245]	1.07	19.94	18.58	2.67
every other readout	0.298 [.243, .359]	0.97	19.72	18.55	2.95
never (SPT-FLOW)	0.332 [.306, .376]	1.05 [.78, 1.19]	19.82	18.42	2.03

Table 5: Each variant changes one setting relative to the reference: $\beta = 3 \times 10^{-3}$, latent dimension 32, and 8 Euler steps during training. Variants use one seed; the reference reports three-seed means and the specificity range. PSNR is in dB.

Configuration	mid best-of-10	mid mean	final best-of-10	div.	specificity
$\beta = 10^{-3}$	19.79	18.83	18.36	1.56	0.44
reference ($\beta = 3 \times 10^{-3}$)	19.82	18.85	18.42	2.03	0.31–0.38
$\beta = 10^{-2}$	19.53	18.82	17.98	1.06	0.20
latent dimension 16	19.81	18.71	**18.64**	2.66	**0.58**
16 Euler steps	**19.85**	**19.13**	18.34	0.95	0.26

4.5 TRAINING BUDGET AND SUPERVISION ABLATIONS

Training budget. At 60k steps, teacher-forced latent conditioning improves mid-trajectory best-of-10 PSNR over ODEWorld by only 0.07 dB, whereas SPT-FLOW retains a 1.13 dB advantage (Table 1). Its diversity decreases from 2.03 to 1.18, and the rollout control accounts for a larger share of the improvement: 0.94 of 1.13 dB, compared with 1.52 of 2.01 dB at 30k steps. At 120k steps, the mid-trajectory advantage over ODEWorld remains 0.97 dB in one seed and 1.53 dB in the other, with final-frame advantages of 2.07 and 2.37 dB (Appendix D). A similar decline occurs on the ten-task subset, with gains of 1.36, 1.15 and 0.84 dB at 10k, 20k and 30k steps (Appendix B). This subset entails greater data reuse but also changes task coverage, so it does not isolate the effect of repeated exposure.

Ground-truth re-anchoring. Reducing access to intermediate ground-truth states increases mean posterior specificity from 0.236 to 0.298 and 0.332 across the rollout variants (Table 4). Only fully free-running training exceeds the 0.25 threshold in all three seeds. Nevertheless, re-anchoring at every readout already yields a 1.07 dB swap gap, indicating latent utilization even below this threshold. The swap gap does not increase monotonically with specificity, and the most frequently anchored rollout achieves the highest PSNR. These results distinguish posterior variation, latent utilization, and prediction accuracy. Because the rollout variants share their posterior configuration, their specificity trend supports the proposed supervision-shortcut interpretation; comparisons with teacher forcing additionally involve the implementation differences documented in Appendix A.

Hyperparameter sensitivity. Mid-trajectory best-of-10 PSNR ranges from 19.53 to 19.85 dB across the tested configurations, while posterior specificity varies more substantially (Table 5). Increasing β from 10^{-3} to 10^{-2} reduces specificity from 0.44 to 0.20, with the swap gap decreasing to 0.38 dB at the higher weight. A 16-dimensional latent achieves the highest final-frame best-of-10 PSNR, whereas 16 Euler steps yield the highest mid-trajectory mean PSNR and lowest diversity. These comparisons are exploratory because each non-reference setting uses one seed. Reducing the effective batch size from 16 to 8 also produces near-zero specificity (0.004–0.005) and zero diversity in both tested seeds, indicating sensitivity to the optimization configuration.

Sampling temperature. Scaling the prior standard deviation trades mean accuracy against best-of-10 accuracy and diversity without retraining (Table 6). Lower temperatures improve mean PSNR but reduce both best-of-10 PSNR and diversity, which falls from 2.52 at $T = 1.3$ to 1.08 at $T = 0.5$. At $T = 0.5$, final-frame best-of-10 PSNR falls slightly below the rollout control (18.12 versus 18.18), whereas $T = 0.7$ already exceeds it (18.25). None of the tested temperatures matches the control's

Table 6: Temperature T scales the prior standard deviation at inference, without retraining; $T = 1$ is the default and SPT-FLOW is trained for 30k steps. PSNR is reported as mean ± standard deviation across three seeds, and diversity as the seed mean.

T	mid best-of-10	mid mean	final best-of-10	final mean	div.
0.5	19.53±.07	**18.95**±.09	18.12±.09	**17.40**±.09	1.08
0.7	19.68±.09	18.91±.09	18.25±.08	17.36±.10	1.48
1.0	19.82±.10	18.85±.10	18.42±.08	17.29±.10	2.03
1.3	**19.91**±.10	18.77±.10	**18.51**±.08	17.21±.11	**2.52**

final-frame mean PSNR of 17.77. Relative to $T = 1$, increasing the temperature to 1.3 improves best-of-10 PSNR by 0.09 dB at both frames while reducing mean PSNR by 0.08 dB. The default therefore represents an intermediate setting within the evaluated range.

KL-based mitigations. The tested KL adjustments do not recover trajectory-associated latent utilization under teacher forcing (Appendix F). KL annealing, a threefold reduction in KL weight, and a fourfold increase in the free-bits floor yield specificity values of 0.010–0.027 and swap gaps no larger than 0.001 dB. These results indicate that the failure is not resolved by the tested regularization settings and are consistent with the supervision-shortcut hypothesis.

5 CONCLUSION

We introduced SPT-FLOW, a stochastic extension of ODEWorld that combines a trajectory-level latent variable with rollout matching while retaining a single physical-time integration per sampled trajectory. On LIBERO, at 30k steps, it improves best-of-10 PSNR over our retrained ODEWorld baseline by 2.0 and 2.4 dB at the mid-trajectory and final frames, respectively, and reduces mean LPIPS by 23%, with comparable per-sample inference latency. The prediction advantage persists when both models receive a fourfold increase in training steps. The rollout control recovers most of the accuracy improvement; learned latent conditioning increases diversity and best-of-K performance, with lower mean sample accuracy relative to that control.

Our results also distinguish predictive accuracy from latent utilization. Under teacher-forced supervision, best-of-K accuracy can improve despite low posterior specificity and no measurable PSNR benefit from trajectory-matched latents. Rollout matching exceeds the predefined specificity threshold across all three seeds, and the corresponding inferred latent outperforms a swapped latent by 1.05 dB. Re-anchoring experiments further associate reduced access to intermediate ground-truth states with higher posterior specificity.

Scope and limitations. These findings motivate reporting latent diagnostics alongside prediction metrics. They support trajectory-associated information use, while leaving same-condition semantic mode coverage unresolved. Evaluation covers held-out trajectories from tasks represented in training; generalization to unseen tasks and downstream control benefits remain untested. Latent utilization also depends on the optimization setting, including effective batch size (Section 4.5). The training comparisons match steps, and the rollout control retains random evaluation inputs, limiting claims about compute efficiency and the isolated contribution of learned conditioning.

REPRODUCIBILITY STATEMENT

Section 4.1 describes the data splits, evaluation protocols, and reporting across random seeds. Appendix A documents data preparation, model architectures, hyperparameters, training procedures, numerical solvers, and implementation differences between comparison models. Diagnostic definitions and thresholds appear in Sections 3.4 and 4.3, with additional analyses in Appendix G. Hyperparameter and sampling-temperature sensitivity analyses are reported in Section 4.5 and Appendix C; extended-budget results appear in Appendix D. Development-set results are explicitly identified in Appendix B. Training and evaluation code is included in the supplementary material and will be publicly released.

AI USE STATEMENT

Generative AI tools assisted with language editing and with reviewing the clarity and scope of manuscript claims. The authors reviewed all AI-assisted revisions and take full responsibility for the final manuscript, including its claims, citations, and experimental reporting.

REFERENCES

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. In *International Conference on Learning Representations (ICLR)*, 2018.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

Christopher M. Bishop. Mixture density networks. Technical Report NCRG/94/004, Aston University, 1994.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, et al. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2025.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, 2016.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations (ICLR)*, 2016. Active-units statistic: Sec. 5.2, threshold justified in App. C.

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 (NeurIPS)*, 2024.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. In *International Conference on Learning Representations (ICLR)*, 2017.

Sheng Cheng, Deqian Kong, Jianwen Xie, Kookjin Lee, Ying Nian Wu, and Yezhou Yang. Latent space energy-based neural ODEs. *arXiv preprint arXiv:2409.03845*, 2024.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion Policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

Silvia Chiappa, Sébastien Racanière, Daan Wierstra, and Shakir Mohamed. Recurrent environment simulators. In *International Conference on Learning Representations (ICLR)*, 2017.

Hien Dang, Tho Tran, Tan Nguyen, and Nhat Ho. Beyond vanilla variational autoencoders: Detecting posterior collapse in conditional and hierarchical variational autoencoders. In *International Conference on Learning Representations (ICLR)*, 2024.

Tal Daniel, Carl Qi, Dan Haramati, Amir Zadeh, Chuan Li, Aviv Tamar, Deepak Pathak, and David Held. Latent particle world models: Self-supervised object-centric stochastic dynamics modeling. In *International Conference on Learning Representations (ICLR)*, 2026.

Jeffrey De Fauw, Sander Dieleman, and Karen Simonyan. Hierarchical autoregressive image models with auxiliary decoders. *arXiv preprint arXiv:1903.04933*, 2019.

Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *PMLR*, pp. 1174–1183, 2018.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural ODEs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1993.

Jean-Yves Franceschi, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. Stochastic latent residual video prediction. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *PMLR*, pp. 89–102, 2020.

Songwei Ge, Aniruddha Mahapatra, Gaurav Parmar, Jun-Yan Zhu, and Jia-Bin Huang. On the content bias in Fréchet video distance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Anirudh Goyal, Alessandro Sordoni, Marc-Alexandre Côté, Nan Rosemary Ke, and Yoshua Bengio. Z-forcing: Training stochastic recurrent networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Pengsheng Guo and Alexander G. Schwing. Variational rectified flow matching. In *International Conference on Machine Learning (ICML)*, 2025.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *PMLR*, pp. 2555–2565, 2019.

Danijar Hafner, Jurgis Pašukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024.

Xixi Hu, Bo Liu, Xingchao Liu, and Qiang Liu. AdaFlow: Imitation learning with variance-adaptive flow-based policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *International Conference on Machine Learning (ICML)*, 2025.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a.

Yuhang Huang, Jiazhao Zhang, Shilong Zou, Xinwang Liu, Ruizhen Hu, and Kai Xu. LaDi-WM: A latent diffusion-based world model for predictive manipulation. In *Conference on Robot Learning (CoRL)*, 2025b.

Ferenc Huszár. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*, 2015.

Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. Free bits: Appendix C.8, Eq. (15).

Alex Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

Xuechen Li, Ting-Kam Leonard Wong, Ricky T. Q. Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

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

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer in lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

Dongxiu Liu, Haoyi Niu, Peng Cheng, Yuan Gao, Xirui Kang, Sangli Teng, Koushil Sreenath, and Xianyuan Zhan. ODEWorld: A continuous predictive architecture via physical-time flow. *arXiv preprint arXiv:2607.27924*, 2026.

Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *International Conference on Learning Representations (ICLR)*, 2016.

Đorđe Miladinović, Kumar Shridhar, Kushal Jain, Max B. Paulus, Joachim M. Buhmann, Mrinmaya Sachan, and Carl Allen. Learning to drop out: An adversarial approach to training sequence VAEs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Alexander Norcliffe, Cristian Bodnar, Ben Day, Jacob Moss, and Pietro Liò. Neural ODE processes. In *International Conference on Learning Representations (ICLR)*, 2021.

Sunghyun Park, Kangyeol Kim, Junsoo Lee, Jaegul Choo, Joonseok Lee, Sookyung Kim, and Edward Choi. Vid-ODE: Continuous-time video generation with neural ordinary differential equation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 2412–2422, 2021.

Francois Porcher, Nicolas Carion, Karteek Alahari, and Shizhe Chen. Flow matching in feature space for stochastic world modeling. *arXiv preprint arXiv:2606.29059*, 2026.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.

Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *PMLR*, 2018.

Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Maximilian Seitzer, Sjoerd van Steenkiste, Thomas Kipf, Klaus Greff, and Mehdi S. M. Sajjadi. DyST: Towards dynamic neural scene representations on real-world videos. In *International Conference on Learning Representations (ICLR)*, 2024.

Rishi Shah and Rishav Shrestha. Rollout-decoded reconstruction for long-horizon prediction in latent world models. *arXiv preprint arXiv:2608.25017*, 2026. Concurrent work.

Viktor Sip, Martin Breyton, Spase Petkoski, and Viktor Jirsa. Double projection for reconstructing dynamical systems: between stochastic and deterministic regimes. *arXiv preprint arXiv:2510.01089*, 2025.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 3483–3491, 2015.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Erik Talvitie. Self-correcting models for model-based reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

Luca Anthony Thiede and Pratik Prabhanjan Brahma. Analyzing the variety loss in the context of probabilistic trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

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

Arun Venkatraman, Martial Hebert, and J. Andrew Bagnell. Improving multi-step prediction of learned time series models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, 2009.

Yixin Wang, David M. Blei, and John P. Cunningham. Posterior collapse and latent variable non-identifiability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

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.

Xi Ye and Guillaume-Alexandre Bilodeau. STDiff: Spatio-temporal diffusion for continuous stochastic video prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Yufei Ye, Maneesh Singh, Abhinav Gupta, and Shubham Tulsiani. Compositional video prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Xuanran Zhai, Qianyou Zhao, Qiaojun Yu, and Ce Hao. VFP: Variational flow-matching policy for multi-modal robot manipulation. *arXiv preprint arXiv:2508.01622*, 2025.

Jianing Zhang, Kefan Jiang, Bohan Dai, Yiwen Lu, Alan Yuille, Yilun Du, and Kaiming Chen. World-in-world: World models in a closed-loop world. In *International Conference on Learning Representations (ICLR)*, 2026. Oral.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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 (ICML)*, 2025.

Yizhou Zhou, Chong Luo, Xiaoyan Sun, Zheng-Jun Zha, and Wenjun Zeng. VAE^2: Preventing posterior collapse of variational video predictions in the wild. *arXiv preprint arXiv:2101.12050*, 2021.

A Implementation details

Encoders and latent. The image encoder is a frozen DINOv2-with-registers backbone at 224×224; the language encoder is frozen. The decoupling module is a four-layer cross-attention block producing a single 768-dimensional latent token. The velocity field is a three-layer MLP of width 4096 with FiLM conditioning on a learned sinusoidal embedding of τ. The posterior and prior are two-layer MLPs of width 1024 and 512 producing a 32-dimensional Gaussian.

Conditioning. We condition on the instruction rather than on a goal image. Supplying the final frame determines most of the continuation from the conditioning alone, which leaves little for a trajectory-level latent to represent; the instruction leaves the choice of continuation open, which is the setting the diagnostics of Section 3.4 are about.

Optimization. AdamW, learning rate 10^{-4} with 500 warm-up steps and cosine decay to zero over the declared step budget, weight decay 10^{-4}, gradient clipping at 1.0, batch of four trajectories with four sampled times each, distributed over four A100 or A800 GPUs (effective batch 16; Section 4.5 shows why this matters). The first 1000 steps train the reconstruction term only. Rollout matching integrates with eight Euler steps between consecutive sampled times; inference uses a fixed-step fourth-order Runge–Kutta solver with 200 evaluations for a 50-frame trajectory.

Because the learning rate anneals over the declared budget, a model taken partway through a 60k-step run is not comparable to the final model of a 30k-step run. Every comparison in the main text is between runs at their own final step; the sweep of Appendix B compares the same runs at matched intermediate steps, which is a comparison across arms at each step and not across steps.

Differences between the teacher-forced and rollout arms. The two arms share the encoder, decoder, latent dimension, KL weight and free-bits floor, but they are not identical in every other respect, and we list the differences here so that the comparison can be read for what it is. In the teacher-forced arm the posterior reads $(z_0, z_\tau^\star, z_g, \tau)$ at the single sampled time and a fresh ϵ is drawn for each sampled time; in the rollout arm it reads all $G = 4$ readout latents through a permutation-invariant pooling and one ϵ is shared by the four readouts. The free-bits floor is applied per dimension in both arms, to the batch-averaged KL in the teacher-forced arm and per trajectory before averaging in the rollout arm. We did not run a teacher-forced arm with the pooled posterior, so the main-text attribution to the supervision rests on the re-anchoring series of Table 4, which keeps the rollout posterior fixed. Our retraining of the base objective also lets gradients flow through the Jacobian-vector-product target, whereas Liu et al. (2026) describe a stop-gradient; the released code ships inference only, so we could not run the original training objective as is.

KL term. $\beta = 3 \times 10^{-3}$ with a per-dimension floor of 0.05 applied before summation. The floor is required (Section 4.5); the value was selected on the development half.

Dataset. LIBERO demonstrations are re-rendered at 256×256 by restoring each stored simulator state and rendering offscreen, rather than by replaying actions, so the rendered trajectory matches the recorded one exactly. This yields $5{,}000$ trajectories over 100 tasks. The held-out set contains every tenth demonstration of each task.

Evaluation. PSNR, LPIPS (Zhang et al., 2018) and SSIM (Wang et al., 2004) are computed on decoded frames at a mid-trajectory index and at the final index. Best-of-K and the mean over K samples follow the protocol of stochastic video prediction (Babaeizadeh et al., 2018; Denton & Fergus, 2018), and diversity is the mean pairwise distance between samples as in Lee et al. (2018), taken here between terminal latents. FVD (Unterthiner et al., 2018) uses a standard I3D feature extractor on 16-frame clips and is biased toward per-frame quality over temporal realism (Ge et al., 2024), so we read it as one number of three rather than a summary. SSIM moves with LPIPS and is omitted from Table 3 for space: over the same three seed pairings, ODEWorld 0.743, rollout training without the latent 0.789 and SPT-FLOW 0.792 (best of K) at the mid-trajectory frame, and 0.714, 0.774 and 0.781 at the final frame. Perceptual and distributional metrics use at least 50 trajectories: on ten trajectories we measured differences that disappeared at fifty, in both directions, so smaller samples are not reported.

Per-step baseline. An untuned per-step flow-matching world model (Lipman et al., 2023; Porcher et al., 2026) trained on the same encoder and data reaches 13.5 dB at the mid-trajectory frame with 500 network evaluations, and doubling its solver steps changes this by only 0.13 dB; further tuning might narrow the gap, and we do not report it in the main tables.

B CROSS-SCALE CHECK ON TEN TASKS

Table 7: A 10-task subset (450 training trajectories): the same runs at matched intermediate steps, one seed, mid-trajectory best-of-10 on that subset's 40-trajectory development split (not the test split used elsewhere).

10 tasks	10k	20k	30k
ODEWorld	18.37	18.85	18.98
+ latent, tf	18.44	19.28	19.45
SPT-FLOW	**19.73**	**20.00**	**19.82**

On a 10-task subset (Table 7) SPT-FLOW leads at every step, by 1.36, 1.15 and 0.84 dB. The margin shrinks as training continues, in the same direction as the budget result. These two settings differ by an order of magnitude in passes over the data at equal step counts, which is consistent with the erosion being governed by data repetition rather than by step count, though we have not isolated the two.

C HYPERPARAMETER SENSITIVITY OF THE FIXED METHOD

Table 5 in the main text changes one knob at a time from the reported configuration, and Section 4.5 discusses it. The swap gap of the $\beta = 10^{-2}$ model is 0.38 dB against 1.05 for the reference.

The sampling-temperature sweep is Table 6 in Section 4.5.

D FOUR TIMES THE TRAINING STEPS

Table 8: Seed 0 of each arm trained for 120k steps on a different accelerator (RTX 5090; other tables use A100s, and the reference model reproduces its Table 1 numbers on this node to four decimals), 250-trajectory test split, $K = 10$. A second seed is reported in the text. Models taken partway through share the 120k cosine schedule and are not comparable with the endpoints of Table 1.

	ODEWorld		SPT-FLOW			
Step	mid	final	mid best-of-10	mid mean	final best-of-10	diversity
30k	18.10	15.87	19.45	18.49	18.19	1.56
60k	18.10	15.71	19.88	19.17	18.67	0.98
90k	19.00	16.81	19.85	19.17	18.85	0.83
120k	19.07	16.88	**20.04**	**19.37**	**18.95**	0.68

At the 120k endpoint (Table 8) the gap is 0.97 dB at the mid-trajectory frame and 2.07 dB at the final frame, and the deterministic model's own curve has flattened (19.00 at 90k, 19.07 at 120k). The latent stays trajectory-specific at this budget (specificity 0.31, all 32 dimensions above the threshold; swap gap 0.74 dB against 1.05 at 30k), and the best-of-K slope from $K = 1$ to 30 is 0.75 dB against 1.30 at 30k, so the longer run retains latent utilization while showing a smaller gain from sampling. Its diversity falls from 1.56 to 0.68 over the run, continuing the 2.03 to 1.18 drop between the 30k and 60k rows of Table 1. This comparison does not include a 120k-step rollout control and therefore does not isolate the learned latent's incremental benefit.

A second seed reproduces the endpoint on Ampere hardware: trained on an A100 (SPT-FLOW) and an A800 (ODEWorld), with the reference model reproducing to four decimals on both nodes, the two models reach 20.47 and 18.94 dB at the mid-trajectory frame and 19.37 and 17.00 dB at the final frame, a gap of 1.53 and 2.37 dB against 0.97 and 2.07 for seed 0. The latent remains trajectory-specific at this budget in both seeds (specificity 0.365 and 0.307, all 32 dimensions above the threshold).

E MODE AVERAGING IN A CONTROLLED SETTING

A field trained with first-order supervision on two-mode data integrates to a path between the modes. Trajectories in this task start at a common point and reach a common endpoint along an upper or a lower arc with equal probability. The deterministic field produces neither arc. Adding unstructured Gaussian noise to the field broadens the paths without recovering the two modes in this experiment. A latent-conditioned field recovers both arcs. Wasserstein-1 distance (Villani, 2009) between predicted and demonstrated mid-trajectory positions, over three seeds, is 0.350 for the deterministic field — exactly the arc amplitude, meaning it predicts the midline — 0.143 with additive Gaussian noise, and 0.031 and 0.040 for a latent-conditioned field trained with teacher forcing and with rollout matching respectively. Both latent variants recover the modes here: in this low-dimensional setting, where the posterior reads the whole trajectory, even the teacher-forced latent gets used. Latent neglect is therefore not inevitable under teacher forcing. This positive control verifies mode recovery where modes are known, but does not isolate the effects of scale, posterior inputs, and optimization that differ from the LIBERO setting.

The diagnostics of Section 4.3 read as expected where the latent is known to be used. On 400 fresh trajectories per seed, one of the four latent dimensions carries the mode in every model, with a per-dimension specificity of 0.89–0.98 against the learned conditional prior (mean over dimensions 0.31 ± 0.02 teacher-forced and 0.34 ± 0.04 rollout-trained; the mean is diluted by unused dimensions, so the per-dimension value is the one to compare). The latent swap separates cleanly: the mid-trajectory error of a free rollout is 0.016 and 0.010 with the trajectory's own latent, 0.38 and 0.37 with the latent of a random other trajectory, which lies in the opposite mode about half the time (0.70, the full arc separation, when it does), and 0.37 and 0.33 with a prior draw. Both diagnostics detect mode-associated information in this positive control. Their low readings for the teacher-forced LIBERO model remain specific to that model and evaluation setting. At scale the per-dimension reading agrees with the mean: no dimension of the teacher-forced posterior exceeds 0.022 in any seed, and under rollout training all 32 dimensions exceed 0.25 in every seed.

F CLASSICAL MITIGATIONS IN DETAIL

Annealing the KL weight from zero over the first 5k steps leaves the teacher-forced posterior at specificity 0.014 and 0.012 across two seeds, against 0.007–0.012 without annealing and 0.31–0.38 for rollout training. The free-bits floor is load-bearing in the other direction: lowering it from 0.05 to 0.002 or to zero collapses the KL to 0.014 and 10^{-4}, and the low-floor row of Table 2 shows the resulting specificity. The tested KL schedules and floors do not recover trajectory-associated latent utilization in these teacher-forced configurations.

The additional KL settings in Table 9 yield the same qualitative pattern. Lowering the KL weight by a factor of three or raising the free-bits floor fourfold moves mid-trajectory accuracy by at most 0.20 dB from the reference configuration and leaves specificity between 0.010 and 0.027, with the latent of a trajectory's own future scoring within 0.001 dB of a shuffled one in all three configurations.



Figure 4: Four teleoperated demonstrations of one instruction (*open a drawer and put a snack in the bowl*) on a real arm, one per row; fixed camera except the wrist-camera grasp column. All start with both drawers closed and diverge in which drawer is opened and which snack is taken. t/T is normalized time. Recorded in separate sessions; an illustration of the premise, not an evaluation.

These results are consistent with the supervision-shortcut hypothesis, but do not isolate it from the other implementation differences in Appendix A.

Table 9: Teacher-forced training with one KL knob changed; two GPUs with twice the per-GPU batch (effective batch 16) on the node of Appendix D, 30k steps, seed 0, 250-trajectory test split, $K = 10$. The reference row is retrained on the same node, so rows are comparable with each other but not with Table 1.

Teacher-forced configuration	mid best-of-10	mid mean	specificity	max. dim.	swap gap
reference ($\beta = 3 \times 10^{-3}$, floor 0.05)	18.52	18.19	0.016	0.026	0.001
$\beta = 10^{-3}$	18.67	18.34	0.010	0.015	0.000
free-bits floor 0.2	18.72	18.34	0.027	0.045	0.001

G DIAGNOSTICS

The rollout control's residual noise. The control is trained at $\epsilon = 0$ but evaluated by the same code as the full model, which draws ϵ from a prior the control never trained. Residual sensitivity to this input produces its diversity of 0.51 in Table 1 and allows best-of-K gains. This control does not measure strictly deterministic rollout performance, and its sample mean cannot be substituted for an evaluation at $\epsilon = 0$. The 0.49 dB mid-trajectory increment in Section 4.2 compares best-of-10 against best-of-10 under the reported protocol. Per seed, a paired bootstrap (Efron & Tibshirani, 1993) over the 250 test trajectories gives increments of $+0.25$, $+0.57$ and $+0.63$ dB, each with a 95% interval excluding zero; at the final frame one seed's interval includes zero ($+0.05$, $[-0.10, +0.21]$). These intervals describe trajectory resampling within each fitted seed; they do not establish robustness across unseen tasks or additional training seeds.

Detail behind Table 2. The per-dimension results agree with the aggregate diagnostic: teacher-forced posterior means vary little, whereas under rollout training every one of the 32 dimensions exceeds 0.29 in every seed. Thus, the latter result does not rest on a few active dimensions. On the ten-task subset of Appendix B, the rollout posterior reaches specificity 0.47; this result alone does not identify the information represented by the latent.

Exploratory association between task dispersion and prediction gain. We measured per-task branching as the mean pairwise distance between demonstrations' proprioceptive states at mid-trajectory, resampled to a common time grid, over 12 demonstrations for each of the 100 tasks, and correlated it with the full model's gain over the reported rollout control per test trajectory. Over the 250-trajectory test split the Spearman correlation is 0.005 ($p = 0.94$) at the mid-trajectory frame and 0.115 ($p = 0.07$) at the final frame. On a 60-trajectory subset the same statistic read 0.22, which is why we report the larger sample. These pooled estimates are exploratory: trajectories share task-level dispersion values, and the reported analysis does not establish task-level inferential uncertainty. The small correlations neither establish zero association nor distinguish semantic continuation selection from other trajectory-associated variation. More generally, prediction quality and downstream utility require separate evaluation (Zhang et al., 2026).

Posterior specificity is $\mathrm{std}_n[\mu_q(x_n)]/\,\mathrm{mean}[\sigma_p]$, where μ_q is the posterior mean for held-out trajectory n and σ_p the prior standard deviation, averaged over latent dimensions. Both diagnostics also record whether two distinct latents change the rollout at all, which distinguishes an uninformative posterior from a latent that is not wired into the field.

Qualitative rollouts. Figure 5 shows one test trajectory decoded from every arm of Table 1. The three teacher-forced samples are visually identical, which is the decorative latent of Section 4.3 seen directly; the three SPT-FLOW samples differ from one another in the arm's path and follow the recorded continuation more closely than the deterministic model. The trajectory was chosen as one of the highest-diversity test trajectories for SPT-FLOW, so the spread shown is toward the upper end of what the model produces. Animated versions of four such trajectories are in the supplementary material.

The recorded continuations separate in the latent once the branch is taken. We asked whether the frozen encoder's latent separates the four continuations of Figure 4 over time, encoding 60 time-normalized frames per episode from the fixed camera and subtracting each episode's first-frame latent. Within one recording session, a leave-one-episode-out nearest-centroid classifier cannot tell the two drawers apart reliably from the first frame (0.54, near chance) and then identifies the drawer with accuracy 0.92 from $t/T = 0.05$ and 1.00 from $t/T = 0.15$. This classifier therefore distinguishes the recorded drawer choices more accurately after branching; its first-frame result does not establish that the representation contains no predictive information. Each continuation was recorded as a block: the same classifier also identifies the snack within a drawer (0.71–1.00) and the session within a continuation (0.61–1.00), and the centroid distance between two sessions of the same continuation (2.2–3.7) matches the distance between drawers within one session (3.2), so session and operator style travel with the choice. Interleaving the orders within one session would separate them.

Time resolution. In the earlier 128-pixel configuration we read the trajectory out at $0.5\times$, $1\times$, $2\times$ and $4\times$ the training resolution; PSNR was 18.1, 18.05, 18.05 and 18.05 dB at the mid-trajectory frame and 16.77 at the final frame in all four cases. This checks prediction consistency at the evaluated frames across readout resolutions; it does not measure the cost of decoding additional frames.

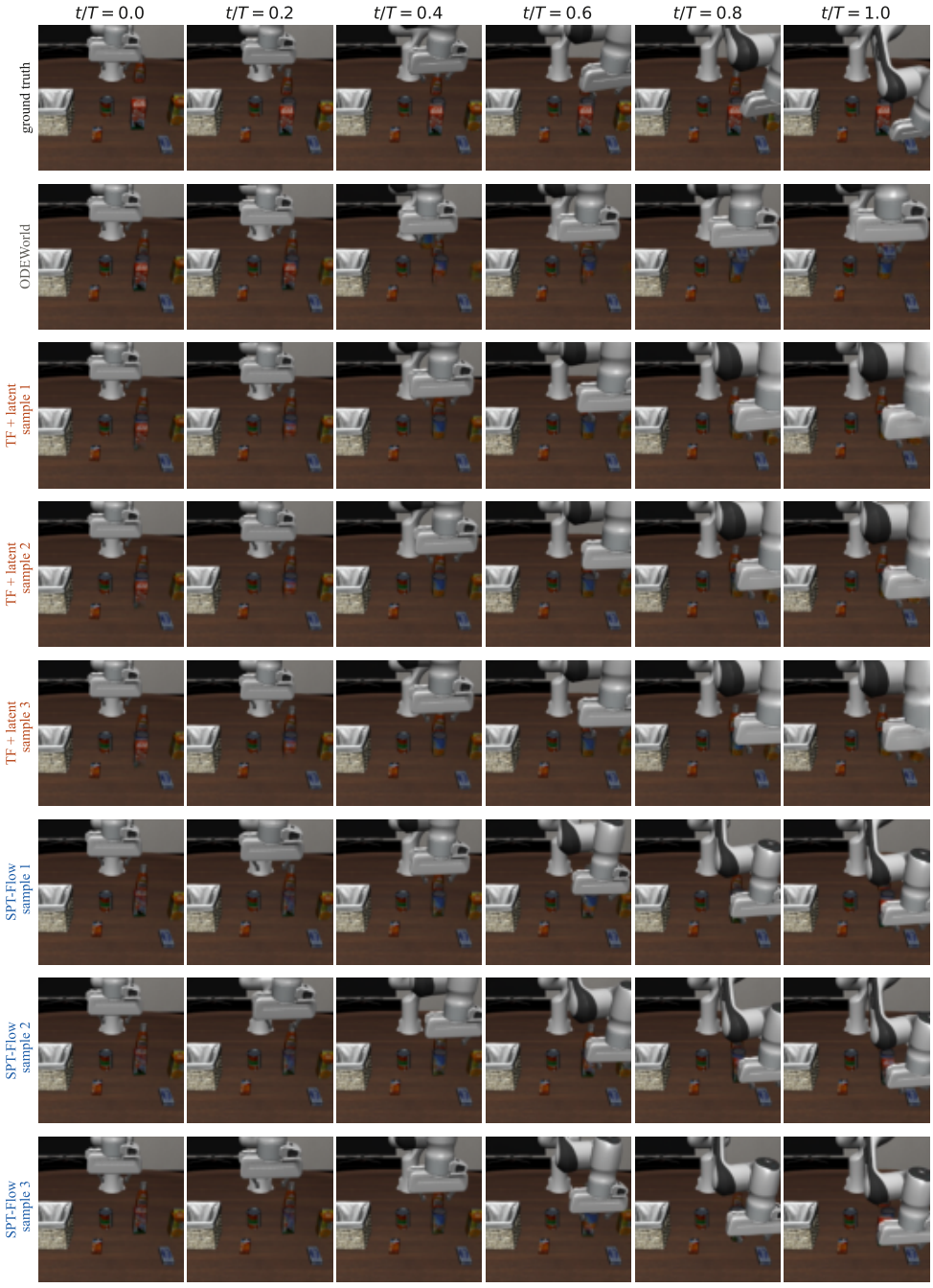

Figure 5: One test trajectory (*put both the cream cheese box and the butter in the basket*), six of 50 frames, decoded through the shared frozen decoder. Rows: recorded continuation, ODEWorld, three samples from the teacher-forced latent model, three samples from SPT-FLOW; all seed-0 models at 30k steps from Table 1.