ACTION JETS: LEARNING THE LOCAL SHAPE OF THE ACTION-CONDITIONED DATA MANIFOLD

Anonymous authors
Paper under double-blind review

ABSTRACT

World models are a powerful means of reasoning and planning for a variety of applications including robotics. However, many of these world models such as video generation models are high dimensional and computationally expensive which limits the number of action candidates that can be evaluated by a planner. We show that this cost is avoidable for planners over short horizons which only local information with respect to the current state. For a fixed state, the frames reachable by varying a low-dimensional action, a camera movement or an end-effector command, lie on a low-dimensional manifold. We estimate the local shape of that manifold directly via a network trained to estimate how each image region moves per unit of each action axis supervised by data collected from samples by the generator. A correction head fit to trajectory data estimates the shape of this manifold region up to a third-order Taylor polynomial. We call this network estimating the Taylor polynomial describing the manifold shape an *action jet* and compute its radius of validity in closed form. By traversing this approximation of the local manifold shape, we only have to perform inference once per state for any number of candidate actions. We achieve over 100x speedup of same length trajectory rollouts compared to the source generators, and this gap widens with the number of candidates a planner evaluates. We measure against state-of-the-art action-conditioned generators: IRASim and WorldGym on BridgeData V2, a real 7-DoF manipulation benchmark, and DepthSplat, LagerNVS and FrameCrafter on DL3DV scenes. Because only the first-order term comes from the generator and the rest from the data, the action jet also predicts held-out frames more accurately than IRASim, the generator that supplied its tangent, at short horizons and at roughly 10^4x less compute.

1 INTRODUCTION

Robot systems increasingly use such generators as world models, to score candidate actions, to evaluate policies offline, and to plan (Zhu et al., 2025; Quevedo et al., 2026; Yang et al., 2024; Bar et al., 2025). Every prediction requires a full image generation, so a planner comparing a hundred candidates performs a hundred rollouts. IRASim (Zhu et al., 2025) spends 25 diffusion steps on each candidate action chunk and WorldGym (Quevedo et al., 2026) 40, so this scale is a significant barrier to real time decision making.

However, planners over short horizons only require local information with respect to the current state. Fix the state x of a scene and vary an action a, such as a camera displacement or a robot end-effector command. The resulting latent image states trace a surface $\mathcal{M}_x$ (Fig. 1a), the action-conditioned data manifold. This manifold is a property of the data distribution, including the scenes, cameras and dynamics, and not of any specific model. Every action-conditioned generative model trained on that distribution carries an approximation of $\mathcal{M}_x$ that can be reached only by sampling.

This paper explicitly estimates the local shape of action-conditioned data manifolds. A *chart* is a homeomorphism between a subset of a topological manifold and a subset of $\mathbb{R}^n$, where n is the dimension of the manifold. Freezing a generator T and fixing the input state gives a map from actions to latent future states, specifically a chart of $\mathcal{M}_x$ (Fig. 1b). We differentiate it with per axis action probes to measure how each image region changes per unit of each action axis. The probe returns a field: one vector for every image token, saying how that token moves per unit of

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.

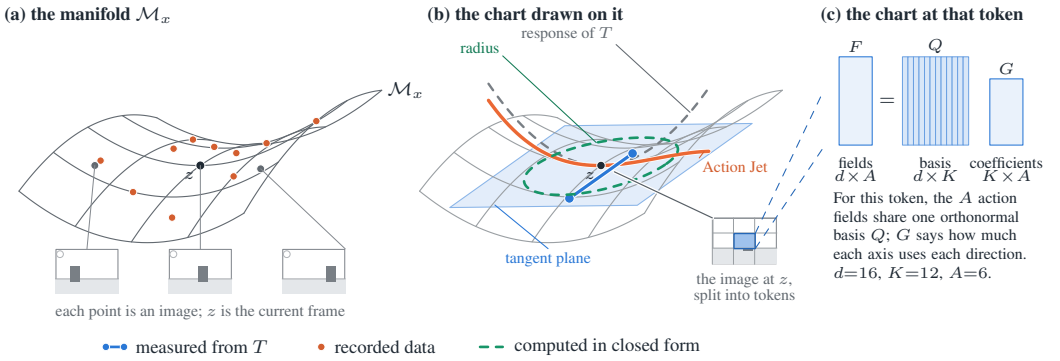

Figure 1: **A chart of the action-conditioned data manifold.** *(a)* For a fixed state x, varying the action traces a curved surface in latent space, the manifold $\mathcal{M}_x$, on which transitions recorded from x lie. *(b)* Probing the frozen generator T at $\pm\delta$ (blue) gives the tangent plane at the latent state z. While T's own response bends away from the manifold, the action jet (orange) fits the Taylor polynomial to the recorded transitions in the data and so follows $\mathcal{M}_x$ up to the radius r (green) it computes from its own coefficients. *(c)* One token of the chart is marked on the tangent plane in (b). The geometry encoder writes the fields of all A action axes in one orthonormal basis Q with coefficients G.

the action axis. We refer to this result as the *action Jacobian* for each image token. A geometry encoder learns to emit the span of that action Jacobian, an orthonormal tangent basis with per-axis coefficients, in one forward pass (Fig. 1c). A correction head adds the second- and third-order terms of the local Taylor expansion as coefficient fields of a polynomial in the action space, trained on recorded transitions (Fig. 2). We compute from this Taylor polynomial the radius of validity for our approximated local shape or chart of the manifold. A *jet* is an operation that takes a differentiable function and produces a Taylor polynomial at each point of its domain. We call our approximation method of the Taylor polynomial describing the local shape of the action-conditioned data manifold an *Action Jet*.

Contributions and significance. This paper introduces a method that estimates the local shape, specifically a chart, of action-conditioned data manifolds. By performing the local approximation of this manifold, we enable significant practical improvement for efficient and accurate candidate trajectory generation of image-action pairs. Instead of having to perform inference with the generative model for every action candidate, we only have to perform inference with the generative model once per state for any number of candidate actions. After sampling the manifold with the generator, we can access specific data points within the radius of validity of the associated action jet via matrix multiplication alone. This difference results in an over 100x speedup of trajectory rollouts compared to source generative models, and the speed gap scales with the number of evaluated candidates. Furthermore, because only the first order polynomial term comes from the generator and the remaining terms come from the data itself, we can achieve more accurate future state predictions than the trained generator over short horizons.

2 Related Work

Measuring the manifold hypothesis. The hypothesis can be tested statistically (Fefferman et al., 2016) and studied more specifically for image data. The intrinsic dimension of image data has been estimated directly (Pope et al., 2021; Brown et al., 2023), from estimating pose-indexed appearance manifolds (Murase & Nayar, 1995), and from pretrained diffusion models (Stanczuk et al., 2024; Kamkari et al., 2024). Local polynomials have been shown to recover tangents and curvature from samples (Aamari & Levrard, 2019; Philipona et al., 2003). We extend this research on manifold estimation to action-conditioned manifolds.

Geometry of generative models. Pullback metrics describe the curvature of a generator's latent space (Arvanitidis et al., 2018; Shao et al., 2018; Wang & Ponce, 2021), and Jacobian and Hessian structure has been used to find or enforce interpretable directions (Härkönen et al., 2020; Peebles

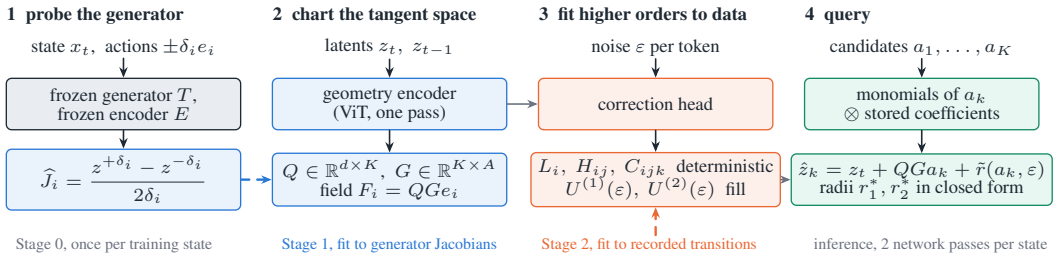

Figure 2: **Method overview.** Fill color marks what supervises each part: blue, the generator's measured Jacobians; orange, recorded transitions; green, no network at all.

et al., 2020; Zhu et al., 2021; Kwon et al., 2023; Park et al., 2023). These analyses concern one model's latent code; we differentiate with respect to a physical action and compare independently trained models.

Jacobians and locally linear control. Visual servoing relates the motion of an image to the motion of a camera through an image Jacobian (Hutchinson et al., 1996), which can be estimated from observed motion without calibration and used inside a trust region (Jägersand et al., 1997), the classical counterpart of our probe and radius. Neural Jacobian Fields learn a dense first-order visuomotor Jacobian per robot (Li et al., 2025b). Latent control models assume locally linear dynamics (Watter et al., 2015; Banijamali et al., 2018) or penalize curvature to keep them locally linear (Levine et al., 2020). Similarly to approaches learning image transformation operators (Rao & Ruderman, 1998; Jayaraman & Grauman, 2015), we estimate the local shape of the manifold directly from the data for a computed radius where our estimate is valid.

Action-conditioned generators. World models predict observations from a state and an action (Ha & Schmidhuber, 2018; Hafner et al., 2025; Hansen et al., 2024), increasingly as video generators conditioned on robot actions (Yang et al., 2024; Bruce et al., 2024; Zhu et al., 2025; Quevedo et al., 2026; Huang et al., 2026; Guo et al., 2026; NVIDIA et al., 2025) or as predictors in pretrained feature spaces (Bar et al., 2025; Zhou et al., 2025; Assran et al., 2025). Camera-controlled video diffusion (Wang et al., 2024; He et al., 2025; Gao et al., 2024; Yu et al., 2025; Wu et al., 2026) and feed-forward novel-view synthesis (Charatan et al., 2024; Chen et al., 2024; Xu et al., 2025; Jin et al., 2025; Wang et al., 2025; Szymanowicz et al., 2026) play the same role for camera motion. Each model has to perform a forward inference pass for each candidate, and model performance is measured by the quality of predicted candidate trajectories (Kang et al., 2025; Motamed et al., 2026; Duan et al., 2025; Li et al., 2025a). We use these models as instruments and compare their local derivatives, enabling all candidate predictions from each state to be calculated from the result of a single forward inference pass.

Derivatives, distillation and agreement. Training on derivatives goes back to Tangent Prop (Simard et al., 1991) and the manifold tangent classifier (Rifai et al., 2011). Sobolev training (Czarnecki et al., 2017) and Jacobian matching (Srinivas & Fleuret, 2018) distill a teacher's derivatives along with its outputs. Output distillation (Hinton et al., 2015; Salimans & Ho, 2022; Song et al., 2023; Yin et al., 2024; Sauer et al., 2024) trains a student to reproduce its teacher, whose quality then bounds the student. We distill only the first derivative with respect to an action and fit the rest to the data itself. The agreement between independently trained models (Kornblith et al., 2019; Bansal et al., 2021) has provided evidence that the models capture a shared reality (Huh et al., 2024; Kadkhodaie et al., 2024). We apply that reasoning to derivative fields that share a coordinate system by construction.

3 Charting the action-conditioned data manifold

Our approach to estimating future states in the local region $\mathcal{M}_x$ has four steps corresponding to Sections 3.1-3.4 which also correspond to each column of Figure 2. First, we probe the generator to compute the action Jacobian for given input states (Section 3.1). We use the results of this probe to train a geometry encoder to directly emit the span of this action Jacobian. The span of the action Jacobian gives an estimate of the first order Taylor polynomial describing the shape of $\mathcal{M}_x$

(Section 3.2). We then fit the higher orders of the Taylor polynomial to the data itself (Section 3.3). This Taylor polynomial is an approximation of $\mathcal{M}_x$ valid inside a radius within which we can query possible trajectory rollouts (Section 3.4).

Problem Statement. Latent image states z are defined in the space of a frozen image autoencoder (E, D) (Rombach et al., 2022): $z = E(x)$. We split z into N tokens, each a 2×2 block of latent pixels, so every token is a vector in $\mathbb{R}^d$. For an image state x, let $\mathcal{M}_x$ be the set of latent image states reachable from x by actions in a neighborhood U of zero. This neighborhood is a smooth surface of dimension at most A, fixed by the data distribution including dynamics. A generator T trained on the dataset defines $\varphi_x(a) = E(T(x, a))$ where and to what extent it has learned the data, $\varphi_x(U) \approx \mathcal{M}_x$. $\varphi_x(U)$ is an approximate chart of the action-conditioned data manifold. The local shape of $\mathcal{M}_x$ is the Taylor expansion

$$\varphi_x(a) \;=\; \varphi_x(0) + \sum_i a_i J_i + \tfrac{1}{2}\sum_{i,j} a_i a_j H_{ij} + \tfrac{1}{6}\sum_{i,j,k} a_i a_j a_k C_{ijk} + \dots, \tag{1}$$

whose first-order fields $J_i = \partial\varphi_x/\partial a_i$ span the tangent space. We introduce *Action Jets* which are networks that emit a truncation of Eq. 1 from the current frame, with the first order measured from generators and the rest fit to data (Figure 2).

3.1 Probe the generator for the action Jacobian

For each generator, state, and axis i we render at $\pm\delta_i$ along that axis alone, encode both renders, and difference them into the field

$$\widehat{J}_i \;=\; \frac{z^{+\delta_i} - z^{-\delta_i}}{2\delta_i} \;\in\; \mathbb{R}^{N\times d}. \tag{2}$$

$\widehat{J}_i$ is the estimated action Jacobian for axis i, and it says how each token moves per unit of that axis. Since $\widehat{J}_i$ records the generator's response to the action rather than its rendering, the measurement is insensitive to how good the generator's samples look. For stochastic generators, we average over multiple samples to measure an expectation of $\widehat{J}_i$. The probe needs no gradients, so any model that maps a state and an action to an image can be probed, including multi-step diffusion samplers.

3.2 Estimate the first-order Taylor polynomial with the geometry encoder

Probing costs two inference passes from the generator per axis per state which a planner scoring many candidate actions cannot afford. We therefore train a *geometry encoder* that predicts the action Jacobian from the current frame in one forward pass, supervised by the probed fields $\widehat{J}_i$ from Section 3.1.

To construct this encoder, first a ViT reads z_t and one context frame z_{t-1} as N tokens. For each token n it emits a feature vector h_n (used by the correction head, Section 3.3), a basis $B_n \in \mathbb{R}^{d\times K}$ where K is the number of candidate actions, and coefficients $G_n \in \mathbb{R}^{K\times A}$. We orthonormalize the basis by taking Q_n to be the Q factor of the reduced QR decomposition of B_n, so $Q_n^\top Q_n = I_K$ by construction. We then predict how token n moves per unit of axis i as $F_{i,n} = Q_n G_n e_i \in \mathbb{R}^d$. Using this construction, the geometry encoder minimizes

$$\mathcal{L}_{\text{enc}} = \sum_{i=1}^{A} \frac{\|F_i - \widehat{J}_i\|_F^2}{\|\widehat{J}_i\|_F^2} \;+\; \lambda_{\text{cov}} \sum_{i=1}^{A} \frac{\|(I - QQ^\top)\widehat{J}_i\|_F}{\|\widehat{J}_i\|_F}, \tag{3}$$

where norms are taken over all tokens and $(I - QQ^\top)$ acts token-wise as $I - Q_n Q_n^\top$. Normalizing each axis by its own energy keeps axes that cause large image motion from dominating the gradient. The second term penalizes the part of each probed field that lies outside the predicted frame.

Stacking tokens gives $F_i \in \mathbb{R}^{N\times d}$, the encoder's estimate of J_i. We use this estimate to construct the first-order truncation of Eq. 1:

$$\widehat{\varphi}_x^{(1)}(a) \;=\; z_t + \sum_{i=1}^{A} a_i F_i. \tag{4}$$

3.3 Fit higher orders of the Taylor polynomial to the data

The first-order chart of Eq. 4 is a plane and $\mathcal{M}_x$ is a curved surface. We add the second- and third-order terms of Eq. 1 with a *correction head*. Unlike the geometry encoder, it is not supervised by the generator; it is trained on recorded transitions, so every term above first order is fit to the data itself.

For each token n the correction head reads the encoder feature h_n and a noise draw ε_n and emits coefficient fields for the polynomial

$$r(a,\varepsilon) \;=\; \sum_i a_i L_i + \tfrac{1}{2}\sum_{i,j} a_i a_j H_{ij} + \tfrac{1}{6}\sum_{i,j,k} a_i a_j a_k C_{ijk} + \sum_i a_i U_i^{(1)}(\varepsilon) + \sum_{i,j} a_i a_j U_{ij}^{(2)}(\varepsilon), \quad (5)$$

where every field lies in $\mathbb{R}^{N\times d}$. H_{ij} and C_{ijk} are the head's estimates of the second- and third-order coefficients of Eq. 1, L_i corrects the first-order term, and $U^{(1)}, U^{(2)}$ are noise-conditioned *fill* fields for disoccluded content. The prediction for action a is

$$\hat{z}(a,\varepsilon) \;=\; \widehat{\varphi}_x^{(1)}(a) + \tilde{r}(a,\varepsilon) \;=\; z_t + \sum_i a_i F_i + \tilde{r}(a,\varepsilon), \quad (6)$$

with $\tilde{r}(a,\varepsilon) \coloneqq r(a,\varepsilon) - r(0,\varepsilon)$ which guarantees $\hat{z}(0,\varepsilon) = z_t$. The correction head is trained with the encoder frozen on recorded trajectories.

3.4 Query candidate future trajectories in the radius of validity

Our estimate of the local manifold does not hold for actions associated with too large state transitions. Conceptually, this idea can be illustrated simply by camera movement. If a camera is moved to a place where there is no visual overlap with the current scene, it is impossible to predict the contents of the new view given the current view. Computationally, the Taylor polynomial estimating the local manifold shape is no longer accurate too far from its expansion point. Because Eq. 5 is an explicit polynomial in a, this *radius of validity* can be computed from the chart's own coefficients.

Let u be a unit direction in normalized action units. The predicted displacement from z_t at step length r is exactly a cubic $P(r) \;=\; \hat{z}(ru) - z_t \;=\; \alpha r + \beta r^2 + \gamma r^3$ with $\alpha = \sum_i u_i(F_i + L_i)$, $\beta = \tfrac{1}{2}H[u,u] = \tfrac{1}{2}\sum_{i,j} u_i u_j H_{ij}$ and $\gamma = \tfrac{1}{6}C[u,u,u]$. With $p_1 = \|\alpha\|$, $p_2 = \|\beta\|$ and $p_3 = \|\gamma\|$, the *radius of validity* is $r^* = \left(-p_2 + \sqrt{p_2^2 + 4p_1p_3}\right)/(2p_3)$, the positive root of $p_2 r^2 + p_3 r^3 = p_1 r$, past which the nonlinear terms can be as large as the linear one. Inside the bound of r^*, any additional candidate actions can be simulated from the current state. Outside that radius, a new chart needs to be approximated.

4 Results

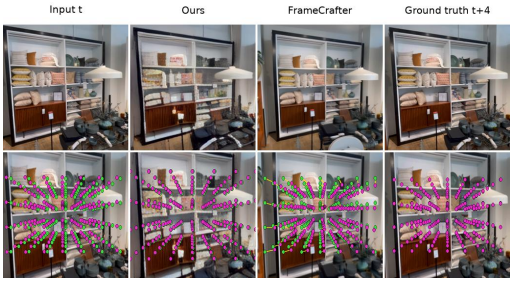

Figure 3: Top: Predictions on DL3DV Benchmark images at g=4. Bottom: points projected through the true (magenta) and recovered (green) camera poses, with their offset being the VCRE.

method	g=1 rot. (deg)↓	g=1 VCRE (px)↓	g=4 rot. (deg)↓	g=4 VCRE (px)↓
GT frame vs. itself (instrument floor)	0.12	0.4	0.10	0.4
Copy input frame (control)	0.39	3.7	1.73	19.1
DepthSplat	0.28	2.2	0.29	2.8
LagerNVS	0.19	1.3	0.27	1.8
FrameCrafter	0.89	8.0	2.16	51.1
Ours	**0.26**	**1.6**	**0.79**	**5.9**

Table 1: Geometric consistency: median rotation error and VCRE (px) of the camera recovered between each prediction and the real frame at its target pose. Action Jets outperform the generative peer at a fraction of inference cost.

	PSNR ↑ (gap 1/2/3/4)				LPIPS ↓ (gap 1/2/3/4)			
tangent only	30.43	26.48	23.66	22.80	.0241	.0486	.0720	.0926
free-form Transformer corrector	34.45	31.19	28.28	27.26	.0153	.0267	.0391	.0511
3-jet corrector (ours)	31.80	29.05	26.92	26.27	.0183	.0305	.0403	.0501

Table 2: Kubric 4-frame horizon prediction through the shared fine-tuned decoder. The free-form corrector re-runs a Transformer network for every candidate action, while the 3-jet answers any candidate by arithmetic on one extraction. The 3-jet reaches LPIPS parity at gaps 3–4 and trails mildly in PSNR, so the polynomial constraint costs little. The 3-jet dominates the 2-jet on all results, and decoder fine-tuning is a strong rendering lever applied to all variants (App. Table 23).

4.1 Experimental setup

We evaluate Action Jet on three different domains. *Kubric* synthetic scenes (Greff et al., 2022) use three action axes (camera azimuth, object translation, light azimuth) and a differentiable generator, so every instrument can be certified against analytic Jacobians. *BridgeData V2* robot manipulation (Walke et al., 2023) has two independently trained action-conditioned world models, IRASim (Zhu et al., 2025) and WorldGym (Quevedo et al., 2026), and we use it to measure where the learned chart holds and its downstream utility. *DL3DV* (Ling et al., 2024), showing camera motion in static environments, has two architecturally unrelated novel-view generators, DepthSplat, built on explicit Gaussian splatting (Xu et al., 2025) with depth estimation, and LagerNVS, a transformer initialized from VGGT (Szymanowicz et al., 2026; Wang et al., 2025). We also use FrameCrafter (Wu et al., 2026) as a diffusion-based generative baseline. DL3DV is used to show that the extracted object belongs to the data rather than the models we probed.

4.2 Kubric: certifying the instrument

Kubric supplies the only setting that provides a differentiable renderer with three declared generative factors (camera azimuth, object translation, light azimuth), so the quantity Eq. 2 estimates exists in closed form. We train a feed-forward next-frame transformer on these scenes and use it as the generator.

Probe-extracted fields match the generator's autodiff Jacobian at cosine $0.87/0.99/0.90$ per factor at δ=0.02, and the encoder reaches held-out per-axis cosines $0.75/0.74/0.78$. Because all three axes share one rank-K basis (§3.2), the intrinsic action dimension is measurable, and the effective rank of the target energy in that basis is 3.9 of K=12.

Table 2 evaluates the 3-jet architecture (§3.3) against tangent-only prediction, and an unconstrained Transformer corrector that runs every query, and so forfeits exactly the property the jet exists to preserve. Free-form Transformer corrector buys 1.0–2.6 dB, shrinking with the gap, and is at LPIPS parity with the 3-jet by gap 3. The jet's advantage over the tangent-only chart, by contrast, grows with the gap, from $+1.4$ to $+3.5$ dB, which is the division of labor of §3.3 showing that the further the step, the more of the displacement lives above first order. Per-factor disentanglement, the holonomy probe, and the ablation that shows sharpening a generator improves its renders and corrupts its derivatives are in App. B.3.

4.3 BridgeData V2: where the chart holds and its utility

BridgeData V2 is a table-top manipulation dataset where a 7-DoF arm moves against a mostly static scene. Two independently trained action-conditioned world models cover it, IRASim and WorldGym, each probed in its own action convention, and we learn a chart from each.

The two generators turn out to define almost unrelated tangent structures on this dataset where per-axis $\cos(\tau_A, \tau_B) = -0.04$ to 0.05, against a ceiling of at least 0.25–0.67 that two generators sharing a signal would have to reach, so the disagreement is not attenuation by sampling noise, and the charts inherit it ($\cos(F_A, F_B) = -0.09$ to 0.12). App. B.5.3 gives the reliability bounds, the convention-invariant subspace comparison, and the held-out anchoring test that identifies that the chart learned from IRASim belongs to the data, which is the model we run evaluations on. The WorldGym-basis chart is kept as a negative control, verifying that it gets worse with breadth, and as a maximally dissimilar chart on which to repeat the radius measurement.

	PSNR ↑				LPIPS ↓				policy	
gap g	1	2	3	4	1	2	3	4	ADE ↓	FDE ↓
copy-frame (control)	24.48	21.48	20.04	19.19	**.051**	**.084**	**.105**	**.121**	1.577	1.994
WorldGym	23.80	21.10	19.70	18.86	.059	.093	.118	.137	1.448	1.783
IRASim (source generator)	23.34	22.37	**21.64**	**21.12**	.101	.116	.127	.136	**0.846**	**1.035**
Action Jet	24.21	22.07	20.67	19.67	.074	.110	.144	.177	1.474	1.992
+ fine-tuned decoder	**24.57**	**22.42**	21.07	20.11	.073	.111	.143	.170	1.299	1.663

Table 3: Open-loop rollouts on BridgeData V2 (n=441), and Octo policy consistency (H=4, a perfect simulator scores 0). The Action Jet basis is charted from IRASim. Action Jet leads on short-horizons against its generator.

Open-loop prediction and policy consistency. We roll out 441 held-out validation episodes under the recorded actions and score each predicted frame against the real one at gaps 1–4 (Table 3). Action Jet with its fine-tuned decoder beats IRASim, the generator that supplied its tangent, at gap 1 on PSNR and LPIPS and is level at gap 2, at roughly $10^4\times$ less compute. From gap 3 IRASim leads by about 1 dB, which is the benefit of video diffusion amortizing the whole trajectory. We also use the Octo policy (Octo Model Team, 2024) to measure policy consistency as ADE and FDE between the action trajectories Octo returns on predicted and on real gap-4 futures under paired sampling keys, so that a perfect simulator scores zero. App. D explains the metric in more detail. This is the one axis where IRASim stays ahead (0.846 against 1.299).

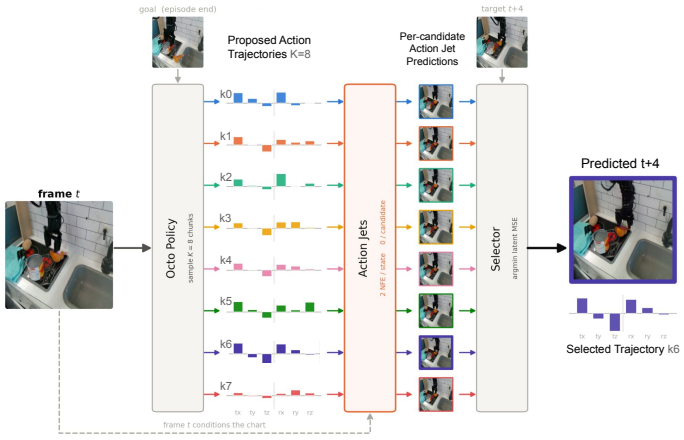

Figure 4: **Parallel Octo policy candidate prediction and evaluation.** Visualization of candidate trajectory samples from Octo being predicted by Action Jet model, with a winning candidate selected based on the lowers latent MSE compared to the true $t+4$ observation. Octo takes goal image as the target produces candidate trajectories 4 steps into the future each planning step, running the planning loop iteratively until the goal state is reached.

cost verifier	ms/cand ↓	K@1s ↑
IRASim	5478	0.2
WorldGym	1094	0.9
Ours	**36.7**	**27**
selected-action MSE ↓ at K		
verifier	4	16	32
random-of-K (control)	.040	.038	.038
oracle-of-K (ceiling)	.006	.003	.003
IRASim	.031	.029	.030
WorldGym	.044	.056	.063
Ours	**.03**	**.028**	**.025**

Table 4: Goal-image candidate verification under a compute budget. K@1s is the number of candidates affordable per second, where Action Jet is the only viable verifier among all baselines. Action Jet is at parity on smaller number of candidates and outperforms all baselines as the number of candidates increases.

Cost. For most World Models, screening each action trajectory candidate costs a full forward pass. Table 5 measures the time to evaluate K candidate four-step futures at one state on identical hardware. Going from one candidate to sixteen multiplies the two diffusion world models inference time by 16.7 and 14.1. Action Jet costs two network evaluations per state regardless of K, so sixteen candidates cost $1.4\times$ one and sixty-four cost $3.1\times$, so the marginal candidate falls to 1.8 ms and the cost per candidate decreases with breadth. Ranking in latent space means the decoder runs once rather than K times. Fig. 4 shows eight candidates on a held-out state, all from a single chart extraction.

		ms/state at K			ms/	K@
model	NFE	1	16	64	cand	1s↑
IRASim	$25K$	5478	91431	—	5478	0.2
WorldGym	$40K$	1240	17501	—	1094	0.9
AJ, composed	8	87.8	604.3	2347.9	36.7	27
AJ, single	**2**	**37.9**	**54.0**	**117.5**	**1.8**	**730**

Table 5: Measured cost in milliseconds of evaluating K candidate four-step futures at one state. Action Jet, even when rolled out autoregressively via composed rollout, is orders of magnitude faster than both diffusion baselines.

	IRASim	WorldGym
copy parity, $\lvert a\rvert/r_1^*$	1.89	2.04
peak gain vs. copy	$+.053$ @1.02	$+.053$ @1.18
1st-order/jet err.	$1.13 \to 0.80$	$1.10 \to 0.70$
$\rho(r_1^*, r_{\text{emp}})$	$+0.61$ $(n{=}28)$	$+0.55$ $(n{=}29)$
median r_{emp}/r_1^*	1.72	2.02
motion realised	0.71	0.63

Table 6: The radius against real Bridge frames for two charts, each in its *own* r_1^* units (96 states, gaps 1–12). "Motion realised", $\|P(a)\|/\|z_{t+g} - z_t\|$, rules out the stationary-frame degeneracy, discussed in §3.4.

Breadth against fidelity. Planning utility is fidelity times the breadth of futures a planner can afford to weigh. Each world model acts as a goal-image verifier over a pool of K candidate action chunks sampled from Octo, predicting every outcome, ranking by latent similarity to the real $t{+}4$ frame, and judging the selected candidate trajectory, with random-of-K and oracle-of-K bracketing every curve (Table 4). Breadth helps competent verification, where every model improves with pool size, and the oracle falls $90\times$ from $K{=}1$ to 128, which shows the benefit of breadth search. Verification skill dissociates from simulation fidelity, shown by WorldGym anti-selecting with its choices worsening from 0.044 to 0.063 as the pool grows, despite mid-pack pixel metrics in Table 3. And under a fixed budget the ranking inverts, since one second buys IRASim a fifth of a candidate while Action Jet evaluates the entire pool. Action Jet wins on both speed and verification.

Validity radius verification. §3.4 computes r_1^* from the corrector's own coefficients, and Bridge lets us check it against real frames since the action from frame i to frame $i{+}g$ grows with the gap, with corresponding ground truth frames available. We sweep over $g = 1 \ldots 12$ horizons, express each step as $|a|/r_1^*(a/|a|)$, which is the demonstrated action in units of the radius the Action Jet predicts along that same direction, and score the decoded prediction in LPIPS against the real frame, alongside the copy baseline, which scores frame i itself against frame $i{+}g$. Measuring the validity radius coverage, we find that 99% of single-step transitions fall inside r_1^* and 67% of four-step ones, the horizon the corrector was trained on. The chart beats the copy baseline out to $1.89\, r_1^*$ and the latent-space error ratio inverts at the same place (Table 6), so a pixel metric and a latent one agree on where the expansion stops. The curves and the per-gap coverage for both charts and the per-state calibration are in App. B.4 (Figs. 7 and 10, Table 9).

Since r_1^* is computed from the IRASim chart, we can repeat the same measurement on the WorldGym chart, which shows nearly orthogonal tangent fields, translation gains a factor of ten apart, a different action convention, and no anchoring on real transitions at all. Both charts nonetheless cross the copy baseline between 1.9 and 2.0 times their own radius, peak at the same advantage, invert the same error ratio, and rank states positively (Table 6). This supports our findings about the validity radius, where r_1^* is computed from the Action Jet alone, at inference cost, and predicts where that Action Jet stops beating the do-nothing baseline, to within a factor stable across two charts that agree about nothing else. Full curves and coverage by gap are in App. B.4.

4.4 DL3DV: TWO GENERATORS, ONE MANIFOLD

DL3DV is a dataset captured by a moving camera in static indoor and outdoor environments, and the action axes are 3D rotations and translations respectively. We use two architecturally distinct generators: DepthSplat which uses explicit Gaussian splatting on estimated depth and LagerNVS which is a transformer initialized from VGGT (Wang et al. (2025)). The hypothesis is that the dataset manifold chart that we learn should be generator-independent.

Forward prediction. Table 7 scores forward prediction, where each baseline is provided with two trailing context frames, against the ground truth target $t{+}g$, on the central 256-pixel window every renderer covers. Among generative methods ours leads FrameCrafter on PSNR and LPIPS at every gap, by $+0.029$ to $+0.222$ LPIPS with every 95% CI excluding zero, at roughly $10^3\times$ less compute. Reconstruction methods re-render observed pixels through explicit geometry, so we report them

	PSNR↑				LPIPS↓			
gap g	1	2	3	4	1	2	3	4
Copy input	18.6	16.0	14.8	14.1	.122	.205	.267	.315
reconstruction (a bound)								
DepthSplat	25.6	23.2	21.1	19.5	.047	.075	.104	.133
LagerNVS	30.5	27.4	25.2	23.7	.039	.062	.088	.115
generative								
FrameCrafter	17.4	15.4	14.3	13.7	.187	.296	.366	.413
Ours	**20.5**	**18.7**	**17.5**	**16.8**	**.159**	**.204**	**.246**	**.279**

Table 7: DL3DV forward prediction, baselines predicting from the two trailing context frames, compared against the ground truth target $t+g$, scored on the central 256-pixel window. A chart distilled from the other generator (DepthSplat) matches this one to within 0.006 LPIPS and 0.09 dB at every gap (App. Table 21).

	t_x	t_y	t_z	r_x	r_y	r_z	mean
DL3DV, n=256, ρ_A=ρ_B=1.000							
learned bases	.77	.81	.77	.84	.82	.79	**.80**
generators	.72	.73	.68	.69	.84	.86	.75
perm. null	.00	−.00	−.00	−.00	.00	−.00	.00
Bridge, n=148							
learned bases	−.02	.06	.08	−.09	.09	.12	**.04**
generators	−.02	.03	.05	−.04	.04	.05	.02
ceiling ≥	.65	.67	.59	.39	.44	.25	.50

Table 8: Tangent agreement $\cos(F_A, F_B)$ of learned bases and $\cos(\tau_A, \tau_B)$ of generators (median, at $0.25\, r_1^*$). Learned bases match and even exceed their generators' agreement where it exists, or inherit its absence (App. B.5.3).

as the bound they are and not as competitors. Against the copy-input control ours wins on PSNR everywhere and on LPIPS at $g \in \{3, 4\}$, ties at g=2 and loses only at g=1.

Geometric consistency. Pixel metrics often cannot distinguish a frame that moved the camera correctly from one that blurred toward the mean, so we score the geometry instead (Table 1, Fig. 3). We pair each prediction with the real frame at the same target pose, match with MASt3R (Leroy et al., 2024), recover the relative camera by PnP, and note that its ground truth is the identity. A frame against itself gives the instrument's noise ($0.12°$, 0.4 px), and the copy control scores the camera motion a predict-nothing model leaves uncorrected. At g=4 Action Jet cuts the control's rotation error by 54% and its reprojection error by 69%, below the control, and beats our generative baseline FrameCrafter. We also provide generators' metrics as the bound and not as competitors.

Two charts, built from unrelated generators, agree. We chart a basis from each generator, probe both at 0.25 of each axis's own r_1^* on 256 identical held-out states, and compare the tangent fields $F = QG$ axis by axis. Each geometry encoder sees only the generator-sampled states, hence F never sees real data frames. The two charts agree at 0.77–0.84 per axis, mean 0.80, more than their generators, which agree at mean 0.75, as shown in Table 8. On four of six axes the learned bases agree with each other more than their generators do, which is what two estimators that each discard the part of their supervision the other does not share would produce. The tangent subspaces coincide at mean principal cosine 0.74 and the induced metrics agree to a relative Frobenius difference of 0.089, so the two charts assign the same lengths as well as the same directions (App. B.5.2). Between BridgeData V2 and DL3DV, the architecture, latent space and downstream real-data supervision are held fixed across the two domains, only the generators' agreement changes, and the bases' agreement changes with it. Thus we empirically validate the hypothesis of charts learning the underlying structure of the action-conditioned data manifold.

5 Conclusion

Action Jets treats the local shape of an action-conditioned data manifold as something to be measured rather than modeled. A single extraction scores any number of candidates in milliseconds, a capability no generative baseline can match, and the fitted curvature beats the source generators at short horizons. Each chart states its validity radius in closed form, and charts built from unrelated generators recover the same tangent structure, evidence that what we extract belongs to the data rather than to one model.

Limitations. All geometry is measured in one fixed autoencoder's latent space, so strictly we recover the composite data-and-encoder manifold, and our experiments vary the generator, not the latent. Disocclusion has no polynomial form, so even with noise injection there is some blur that concentrates at visibility boundaries. Each chart holds only at its probe scale. The advantage over the source generator holds at short horizons. All evaluations are open-loop.

ETHICS STATEMENT

This work studies pretrained generative models and public datasets of robot manipulation and indoor scenes. It involves no human subjects, no personally identifying data and no new data collection. The audit we describe is intended to make the failure modes of action-conditioned generators easier to detect before deployment, and the negative result we report about one publicly described world model is offered in that spirit.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix B give the probing protocol in full, including window derivation, per-axis σ vectors, state selection, noise averaging and the reliability correction. Appendix A states the estimator properties and identifiability conditions with derivations. Every agreement in the paper is reported with its floor, and every stochastic comparison with its ceiling.

AI USE STATEMENT

Generative AI tools were used for manuscript drafting and copy-editing, for producing explanatory diagrams from the authors' own figure specifications, and for searching and verifying related-work citations, as well as polishing code for the release. They were not used for research ideation, experimental design, or analysis; the probing harness, models, experiments and all reported numbers are the authors' own work. The authors reviewed all AI-assisted text and figures, verified every number against the experiment logs that produced it, and take responsibility for the final content of this work.

REFERENCES

Eddie Aamari and Clément Levrard. Nonasymptotic rates for manifold, tangent space and curvature estimation. *The Annals of Statistics*, 47(1):177–204, 2019. doi: 10.1214/18-AOS1685.

Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. In *ICLR*, 2018.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Ershad Banijamali, Rui Shu, Mohammad Ghavamzadeh, Hung Bui, and Ali Ghodsi. Robust locally-linear controllable embedding. In *AISTATS*, volume 84 of *PMLR*, pp. 1751–1759, 2018.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. In *NeurIPS*, pp. 225–236, 2021. URL https://arxiv.org/abs/2106.07682.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *CVPR*, pp. 15791–15801, 2025.

Bradley C. A. Brown, Anthony L. Caterini, Brendan Leigh Ross, Jesse C. Cresswell, and Gabriel Loaiza-Ganem. Verifying the union of manifolds hypothesis for image data. In *ICLR*, 2023. URL https://arxiv.org/abs/2207.02862.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *ICML*, volume 235 of *PMLR*, pp. 4603–4623, 2024.

David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian splats from image pairs for scalable generalizable 3D reconstruction. In *CVPR*, pp. 19457–19467, 2024. doi: 10.1109/CVPR52733.2024.01840.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: Efficient 3D Gaussian splatting from sparse multi-view images. In *ECCV*, 2024. doi: 10.1007/978-3-031-72664-4_21.

Wojciech Marian Czarnecki, Simon Osindero, Max Jaderberg, Grzegorz Świrszcz, and Razvan Pascanu. Sobolev training for neural networks. In *NeurIPS*, pp. 4278–4287, 2017. URL https://arxiv.org/abs/1706.04859.

Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. WorldScore: A unified evaluation benchmark for world generation. In *ICCV*, 2025. URL https://arxiv.org/abs/2504.00983.

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole. CAT3D: Create anything in 3D with multi-view diffusion models. In *NeurIPS*, 2024. URL https://arxiv.org/abs/2405.10314.

Klaus Greff, Francois Belletti, Lucas Beyer, et al. Kubric: A scalable dataset generator. In *CVPR*, 2022.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *ICLR*, 2026. URL https://arxiv.org/abs/2510.10125.

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

Danijar Hafner, Jurgis Pasukonis, 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 *ICLR*, 2024. URL https://arxiv.org/abs/2310.16828.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *NeurIPS*, 2020.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for video diffusion models. In *ICLR*, 2025. URL https://arxiv.org/abs/2404.02101.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. NIPS 2014 Deep Learning Workshop.

Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2World: Crafting video diffusion models to interactive world models. In *ICLR*, 2026. URL https://arxiv.org/abs/2505.14357.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The Platonic representation hypothesis. In *ICML*, volume 235 of *PMLR*, pp. 20617–20642, 2024.

Seth Hutchinson, Gregory D. Hager, and Peter I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, 1996. doi: 10.1109/70.538972.

Martin Jägersand, Olac Fuentes, and Randal Nelson. Experimental evaluation of uncalibrated visual servoing for precision manipulation. In *ICRA*, volume 4, pp. 2874–2880, 1997. URL https://ieeexplore.ieee.org/document/606723/.

Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *ICCV*, pp. 1413–1421, 2015. doi: 10.1109/ICCV.2015.166.

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. LVSM: A large view synthesis model with minimal 3D inductive bias. In *ICLR*, 2025. URL https://arxiv.org/abs/2410.17242.

Zahra Kadkhodaie, Florentin Guth, Eero P. Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *ICLR*, 2024. URL https://arxiv.org/abs/2310.02557.

Hamidreza Kamkari, Brendan Leigh Ross, Rasa Hosseinzadeh, Jesse C. Cresswell, and Gabriel Loaiza-Ganem. A geometric view of data complexity: Efficient local intrinsic dimension estimation with diffusion models. In *NeurIPS*, 2024. URL https://arxiv.org/abs/2406.03537.

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 *ICML*, volume 267 of *PMLR*, pp. 28991–29017, 2025.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *ICML*, volume 97 of *PMLR*, pp. 3519–3529, 2019.

Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. In *ICLR*, 2023. URL https://arxiv.org/abs/2210.10960.

Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, 2024.

Nir Levine, Yinlam Chow, Rui Shu, Ang Li, Mohammad Ghavamzadeh, and Hung Bui. Prediction, consistency, curvature: Representation learning for locally-linear control. In *ICLR*, 2020. URL https://arxiv.org/abs/1909.01506.

Dacheng Li, Yunhao Fang, Yukang Chen, Shuo Yang, Shiyi Cao, Justin Wong, Michael Luo, Xiaolong Wang, Hongxu Yin, Joseph E. Gonzalez, Ion Stoica, Song Han, and Yao Lu. WorldModelBench: Judging video generation models as world models. In *NeurIPS*, 2025a. URL https://arxiv.org/abs/2502.20694.

Sizhe Lester Li, Annan Zhang, Boyuan Chen, Hanna Matusik, Chao Liu, Daniela Rus, and Vincent Sitzmann. Controlling diverse robots by inferring Jacobian fields with deep networks. *Nature*, 643:89–95, 2025b. doi: 10.1038/s41586-025-09170-0.

Lu Ling, Yichen Sheng, Zhi Tu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, 2024.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *WACV*, pp. 948–958, 2026. doi: 10.1109/WACV61042.2026.00099.

Hiroshi Murase and Shree K. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995. doi: 10.1007/BF01421486.

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

Octo Model Team. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems (RSS)*, 2024.

Yong-Hyun Park, Mingi Kwon, Jaewoong Choi, Junghyo Jo, and Youngjung Uh. Understanding the latent space of diffusion models through the lens of Riemannian geometry. In *NeurIPS*, 2023. URL https://arxiv.org/abs/2307.12868.

William Peebles, John Peebles, Jun-Yan Zhu, Alexei A. Efros, and Antonio Torralba. The Hessian penalty: A weak prior for unsupervised disentanglement. In *ECCV*, 2020. doi: 10.1007/978-3-030-58539-6_35.

David Philipona, J. Kevin O'Regan, and Jean-Pierre Nadal. Is there something out there? Inferring space from sensorimotor dependencies. *Neural Computation*, 15(9):2029–2049, 2003. doi: 10.1162/089976603322297278.

Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. In *ICLR*, 2021. URL https://arxiv.org/abs/2104.08894.

Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. WorldGym: World model as an environment for policy evaluation. In *ICLR*, 2026. URL https://arxiv.org/abs/2506.00613.

Rajesh P. N. Rao and Daniel L. Ruderman. Learning Lie groups for invariant visual perception. In *Advances in Neural Information Processing Systems 11 (NIPS 1998)*, pp. 810–816, 1998. URL https://papers.nips.cc/paper/1584-learning-lie-groups-for-invariant-visual-perception.

Salah Rifai, Yann N. Dauphin, Pascal Vincent, Yoshua Bengio, and Xavier Muller. The manifold tangent classifier. In *NeurIPS*, pp. 2294–2302, 2011.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *ECCV*, pp. 87–103, 2024. doi: 10.1007/978-3-031-73016-0_6.

Hang Shao, Abhishek Kumar, and P. Thomas Fletcher. The riemannian geometry of deep generative models. In *CVPR Workshops*, 2018.

Patrice Y. Simard, Bernard Victorri, Yann LeCun, and John S. Denker. Tangent prop – a formalism for specifying selected invariances in an adaptive network. In *Advances in Neural Information Processing Systems 4 (NIPS 1991)*, pp. 895–903, 1991. URL https://proceedings.neurips.cc/paper/1991/hash/65658fde58ab3c2b6e5132a39fae7cb9-Abstract.html.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, volume 202 of *PMLR*, pp. 32211–32252, 2023.

Suraj Srinivas and François Fleuret. Knowledge transfer with Jacobian matching. In *ICML*, volume 80 of *PMLR*, pp. 4730–4738, 2018. URL https://arxiv.org/abs/1803.00443.

Jan Pawel Stanczuk, Georgios Batzolis, Teo Deveney, and Carola-Bibiane Schönlieb. Diffusion models encode the intrinsic dimension of data manifolds. In *ICML*, volume 235 of *PMLR*, pp. 46412–46440, 2024. URL https://proceedings.mlr.press/v235/stanczuk24a.html. arXiv:2212.12611, earlier titled "Your diffusion model secretly knows the dimension of the data manifold".

Stanislaw Szymanowicz, Minghao Chen, Jianyuan Wang, Christian Rupprecht, and Andrea Vedaldi. LagerNVS: Latent geometry for fully neural real-time novel view synthesis. In *CVPR*, 2026. URL https://arxiv.org/abs/2603.20176.

Homer Walke, Kevin Black, Abraham Lee, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

Binxu Wang and Carlos R. Ponce. The geometry of deep generative image models and its applications. In *ICLR*, 2021. URL https://arxiv.org/abs/2101.06006.

Jianyuan Wang, Minghao Chen, Nikita Karaev, et al. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. MotionCtrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, 2024. doi: 10.1145/3641519.3657518.

Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *NeurIPS*, pp. 2746–2754, 2015.

Qi Wu, Khiem Vuong, Minsik Jeon, Srinivasa Narasimhan, and Deva Ramanan. FrameCrafter: Novel view synthesis as video completion. In *ECCV*, 2026. URL https://arxiv.org/abs/2604.08500.

Haofei Xu, Songyou Peng, Fangjinhua Wang, et al. Depthsplat: Connecting gaussian splatting and depth. In *CVPR*, 2025.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *ICLR*, 2024. URL https://arxiv.org/abs/2310.06114.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, pp. 6613–6623, 2024.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. ViewCrafter: Taming video diffusion models for high-fidelity novel view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. doi: 10.1109/TPAMI.2025.3613256.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *ICML*, volume 267 of *PMLR*, pp. 79115–79135, 2025.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. In *ICCV*, pp. 9834–9844, 2025.

Jiapeng Zhu, Ruili Feng, Yujun Shen, Deli Zhao, Zheng-Jun Zha, Jingren Zhou, and Qifeng Chen. Low-rank subspaces in GANs. In *NeurIPS*, pp. 16648–16658, 2021. URL https://arxiv.org/abs/2106.04488.

A FORMAL STATEMENTS

Setup. Fix a frozen autoencoder (E, D) and a state x with latent $z = E(x)$, processed as N blocks so that predictions live in $\mathcal{Z} = \mathbb{R}^{N \times D_b}$. A conditional generator T induces the chart map $\varphi_x : U \subset \mathbb{R}^A \to \mathcal{Z}$, $\varphi_x(a) = E(T(x, a))$, assumed C^3 on a neighborhood of 0 away from visibility events. To the extent T has learned the data, $\varphi_x(U)$ approximates the action-conditioned data manifold $\mathcal{M}_x$, and the tangent fields are $J_i = \partial\varphi_x/\partial a_i|_0$.

Probe accuracy. Taylor expansion with the even orders canceling gives $\widehat{J}_i = J_i + \frac{\delta^2}{6}\partial^3_{iii}\varphi_x + O(\delta^4)$, second-order accurate in δ with all even-order terms removed. For a stochastic generator $T(x, a, \xi)$ the probe estimates $\mathbb{E}_\xi[\widehat{J}_i]$, and averaging M seed-paired renders reduces the noise variance by $1/M$.

Reliability ceiling. Write a measured field as $\widehat{J} = J + \eta$ with noise of relative power $1 - r$, where r is the split-half reliability estimated from seed-paired repetitions and Spearman–Brown corrected to the M-average. For two generators with reliabilities r_A, r_B and true-field cosine c, $\mathbb{E}[\cos(\widehat{J}^A, \widehat{J}^B)] \approx c\sqrt{r_A r_B}$; with one deterministic generator the ceiling is $\sqrt{r_B}$.

Curvature and the ladder. Under the third-order model $\varphi_x(a) = \varphi_x(0) + \sum_i a_i J_i + \frac{1}{2}\sum_{i,j} a_i a_j H_{ij} + \frac{1}{6}\sum_{i,j,k} a_i a_j a_k C_{ijk}$ of Eq. 1, per-unit fields probed at scales $\delta_1 < \delta_2$ along axis i differ by $\widehat{J}_i(\delta_2) - \widehat{J}_i(\delta_1) = \frac{\delta_2^2 - \delta_1^2}{6} C_{iii} + \ldots$, so the measured decorrelation of per-unit fields across probe scales is a direct, axis-resolved image of higher-order structure.

Parity and identifiability. With the tangent supplied by the encoder, the residual $\rho(a) = \varphi_x(a) - \varphi_x(0) - QGa$ expands with an odd deterministic leading term $(J - QG)a$. The linear fields L_i of the curvature head are the parity home of that term; omitting them forces the load through the noise-conditioned fill, where it arrives jittered across hypotheses. Deterministic and stochastic quadratic fields are separated by $\mathbb{E}_\varepsilon[U^{(2)}(\varepsilon)] = 0$, since any mean is absorbed into H, so $(H, U^{(2)})$ are identifiable in expectation.

The identity gauge. Every monomial in $r(a, \varepsilon)$ (Eq. 5) carries at least one factor of a, so the curvature head's polynomial satisfies $r(0, \varepsilon) = 0$; a learned readout bias b violates it. The gauged head $\tilde{r} = r(a, \varepsilon) - r(0, \varepsilon)$ removes any a-independent term exactly while acting as the identity on all monomial terms, hence $\tilde{r}$ equals the polynomial of Eq. 5 whenever the leak is additive.

Single-inference complexity. For A=6: quadratic monomials $n_2 = \binom{A+1}{2} = 21$ and cubic monomials $n_3 = \binom{A+2}{3} = 56$. Scoring K candidate actions at one state costs exactly two network evaluations, plus one fill pass per additional noise hypothesis, and $O(K \cdot N \cdot D_b \cdot (A + n_2 + n_3))$ multiply–adds; no decode is required for latent-space screening.

B EXTENDED RESULTS

This supplement carries the measurements quoted in §4. The instrument ladders and the control inventory that the rest of the supplement refers to are collected at the end, in App. B.7.

B.1 THE METHOD END TO END

The Method's architectural overview is presented in Figure 5, which draws the three parts of the method at the level of tensors. *(A1)* The symmetric probe renders the generator at $\pm\delta$ along one axis and differences the two encodings, so everything the two renders share cancels, and what survives is the generator's response to the action rather than its rendering. *(A2)* The geometry encoder reads the current latent and one context frame and emits an orthonormal basis Q with per-axis coefficients G, so that all A axis fields live in one rank-K subspace instead of being A unrelated predictions, which makes the local dimension of the manifold measurable as an effective rank rather than assumed. *(A3)* The jet corrector emits coefficient fields that depend on the state and a noise draw but never on

the action. This property is what the cost structure of Table 5 rests on, making it so that once the coefficients are extracted, every candidate action is a contraction against tensors already in memory, so the network runs once per state no matter how many candidates are scored, and only the winner is decoded.

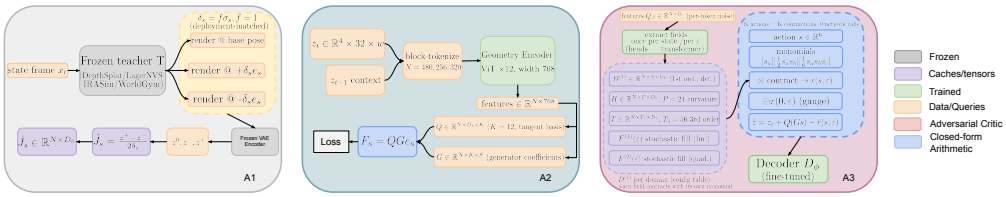

Figure 5: **Architecture.** *(A1)* probing; *(A2)* the geometry encoder, emitting Q and G; *(A3)* the jet corrector, whose coefficient fields depend on the state and a noise draw but never on the action. Everything in the red region is arithmetic on tensors already in memory.

B.2 What a chart emits per-axis

Figure 6 shows what a chart looks like once it has been extracted, on a single held-out state rather than schematically. The state enters the encoder once and the corrector emits its coefficient fields once. Then the six patches of $\mathcal{M}_x$ and the frames traced out on them are read off that single extraction.

Each panel is a two-dimensional patch, not a one-dimensional slice, and that is a property of the drawing rather than of the measurement: a surface needs two coordinates, so each panel sweeps the axis it is labelled with and carries one *companion* axis purely so that there is something to plot against. On the visualized sample of evaluation BridgeData V2 dataset (evaluation slice of the dataset provided by IRASim authors, episode 11), the plotted pairs are (sweep axis, companion axis): $(tx, tz), (ty, tx), (tz, tx), (rx, tx), (ry, tx), (rz, tx)$. Nothing in the figure should be read as a claim about the companion axis because the swept axis is the one whose path carries the markers and whose steps are decoded into the frames next to a surface render.

Two sheets are drawn on every patch. The flat gray one is the first-order chart alone, $z_t + QGa$. It is flat because the plot's x and y are the two action coordinates, so the tangent chart is exactly the $z{=}0$ coordinate plane, parallel to the floor by construction. The blue sheet adds the corrector, $z_t + QGa + r(a)$, and the dashed line off the tangent plane measures how far it departs from the tangent plane, in units of one tangent step length. The two sheets touch at $a{=}0$ but the blue one leaves with a nonzero slope, which is the corrector's linear term L_i.

Every vertex of both sheets is a model evaluation on a grid of commanded actions, and every one of those evaluations is a contraction against already computed coefficient fields. No part of the figure re-runs the network, the encoder pass and the extraction happen once for the whole state, which is the same property that makes the screening cost of Table 5 flat in the number of candidates. Only the frames are decoded, one decode each, which can be done in parallel.

The radius of App. B.4 is the statement that these sheets stay trustworthy out to a computable distance from the origin, and in the rendered surface the validity radius is drawn as a red dashed line. For context, the agreement result of §4.4 is the statement that a chart learned from a different generator produces near-identical bases at the same state: mean per-axis cosine 0.80, subspace principal cosine 0.74, and induced metrics matching to a relative Frobenius difference of 0.089.

B.3 The full instrument check on Kubric

For Kubric, the generator is differentiable, so Eq. 2 can be scored against the quantity it estimates. Per-unit fields match the autodiff Jacobian at cosine 0.87 (camera azimuth), 0.99 (object translation) and 0.90 (light azimuth) at $\delta{=}0.02$. Generator bias, the component of the measured field attributable to the generator rather than the scene, is 0.09–0.16, and the stochastic fill has divergence of 0.017.

Disentanglement and intrinsic dimension. The shared rank-K basis of §3.2 routes different generative factors into different columns, such that off-diagonal generator overlap is 0.19 against 0.38–0.45 measured in the trunk features the basis is read from. The one strong cross-coupling is camera

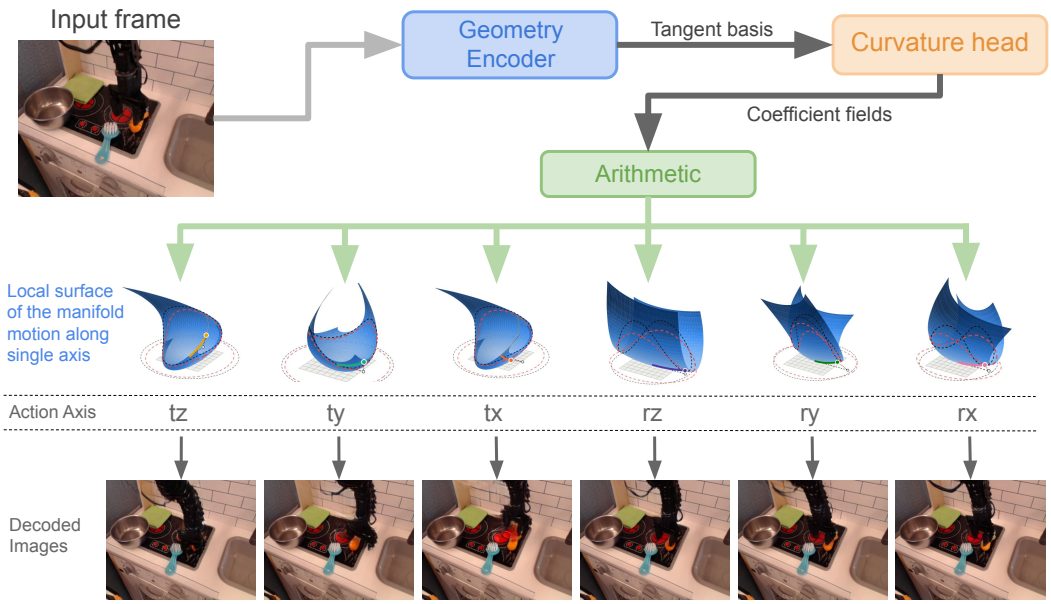

Figure 6: **Extracting and tracing a local chart.** A held-out state is encoded once and the corrector's coefficient fields are extracted once. The six branches then read that one extraction along each action axis. Each branch shows a surface of the local action manifold $\mathcal{M}_x$ and a corresponding decoded frame, predicted from the step along the labeled axis. *Reading a surface render:* x and y are commanded action in the plane spanned by the labeled axis and one companion axis, the latter present only so that the patch is a surface rather than a curve. The dashed circle is the query radius for that axis for that episode, and red dashed circle is the validity radius. The dashed line on the tangent plane is the departure from the tangent basis plane. The flat gray sheet is the first-order chart $z_t + QGa$, which is the $z{=}0$ coordinate plane in this embedding and so is flat by construction. The blue sheet adds the corrector, $z_t + QGa + r(a)$. The two touch at the origin but the blue sheet leaves it with a nonzero slope. Every vertex of both sheets is arithmetic on the single extraction, no network is re-run anywhere in the figure, and only the displayed frames are decoded.

rotation against light rotation at -0.49, which is expected due to high visual similarity. The effective rank of the target energy in the basis is 3.9 of $K{=}12$ against three true action axes, and in-chart coverage is 0.94–0.96.

Non-commutativity. Applying two factors in both orders and comparing the endpoints gives a holonomy that orders by how much the two factors interact. Translation–light pairs score 0.53–0.62 and camera rotation pairs 0.88–1.41. We report the ordering rather than the magnitudes, which are in the units of the loop.

Render quality is not probe quality. This is the experiment behind our generator-selection rule. Adversarially sharpening the generator improved its reconstructions and corrupted its action derivatives, such that probe agreement with the validated reference fell to 0.64–0.68 with $\sim$25% energy inflation. The damage is not confined to the generator alone, since an Action Jet learned from the sharpened generator reproduced it faithfully, reaching per-factor held-out cosines $0.59/0.54/0.61$ against $0.73/0.67/0.77$ for the Action Jet of its unsharpened predecessor. Perceptual losses constrain outputs, not $\partial\,\mathrm{output}/\partial\,\mathrm{action}$, so we select generators by probe agreement and never by sample quality. App. B.5.3 is the same result seen on real data, where the generator with the better pixel metrics (Table 3) is the one whose chart is wrong.

The corrector ladder. Table 23 crosses the corrector variants with the decoder, so the two levers can be read apart. The 3-jet dominates the 2-jet on all sixteen numbers at matched decoder. Decoder fine-tuning is valuable for rendering quality, as for the 3-jet it is worth $+2.7$ to $+3.1$ dB and -43 to -61% LPIPS. And the plain free-form unconstrained Transformer model, which was *not* in the de-

coder's fine-tuning mix, renders best through it, which is the evidence that the corrector comparison in Table 2 is not a decoder artifact.

B.4 THE VALIDITY RADIUS

B.4.1 DEFINITION

Along a unit direction u in action space the Action Jet's prediction displaces the latent by

$$P(r) \;=\; \alpha\, r \;+\; \beta\, r^2 \;+\; \gamma\, r^3, \tag{7}$$

with α collecting the tangent field $F = QG$ and the corrector's linear term, and β, γ the corrector's quadratic and cubic terms. Because the corrector is a cubic in the action by construction, Eq. 7 is exact rather than fitted, and α, β, γ are recovered exactly from five evaluations at $0, \pm\delta, \pm 2\delta$ for any d. Writing $p_1 = \|\alpha\|, p_2 = \|\beta\|, p_3 = \|\gamma\|$, the first-order validity radius solves $p_2 r^2 + p_3 r^3 = p_1 r$:

$$r_1^* \;=\; \frac{-p_2 + \sqrt{p_2^2 + 4 p_1 p_3}}{2 p_3}. \tag{8}$$

It costs six corrector evaluations and no rendering, so it is available per state at inference time. It is a property of the Action Jet alone, requiring no generator and no ground-truth frame. And it is direction-dependent, such that along a demonstrated action a we evaluate Eq. 8 at $u = a/\|a\|$ rather than combining per-axis radii, because the induced metric is strongly anisotropic (App. B.5.2, condition number $\sim$30). r_1^* is invariant to a uniform rescaling of the response, meaning that scaling α, β, γ by a common factor leaves Eq. 8 unchanged, so a chart that simply commands less motion does not thereby report a larger radius. This matters because one of our two Bridge charts under-commands by roughly an order of magnitude. What r_1^* is not invariant to is the higher-order terms collapsing on their own, such as $p_3 \to 0$ at fixed p_1, $r_1^* \to \sqrt{p_1/p_3} \to \infty$, so a chart predicting a nearly stationary frame would report an unbounded radius while predicting nothing, which is the copy-frame failure mode. Every measurement below is guarded by the fraction of the true latent displacement the prediction actually realizes, $\|P(a)\|/\|z_{t+g} - z_t\|$: 0.71 for the IRASim-basis chart and 0.63 for the WorldGym-basis chart, so neither is degenerate and the comparison in App. B.4.3 is between two charts that both move.

B.4.2 MEASURED AGAINST REAL BRIDGE FRAMES

BridgeData V2 is a dataset of real robot arm tabletop manipulation videos, where the action from frame i to frame $i{+}g$ changes with the gap g, and for every g a real frame $i{+}g$ is available. We sweep $g = 1 \ldots 12$ over 96 states and plot everything against $|a|/r_1^*(a/|a|)$.

Coverage. Table 9 gives the share of real transitions inside r_1^*. At gap 1 it is 99%, decaying monotonically to 27% at gap 12, and at gap 4, the horizon the corrector was trained on, it is 67%. Hence the first-order chart covers 99% of single-step Bridge transitions and two thirds of four-step ones.

gap	1	2	3	4	6	8	10	12
IRASim basis	99.0	93.8	76.0	66.7	53.1	39.6	31.2	27.1
WorldGym basis	100.0	90.6	72.9	60.4	41.7	26.0	21.9	22.9
median $\|a\|/r_1^*$ (IRASim)	0.23	0.42	0.59	0.76	0.96	1.16	1.26	1.37

Table 9: Real BridgeData V2 transitions inside r_1^* for both charts by gap. 96 states per gap. The two charts are built from different generators and use different action conventions, yet their coverage curves track each other closely.

The operational question is where prediction stops beating the do-nothing baseline. Table 10 scores the decoded prediction against the real frame in LPIPS, alongside the decoder round-trip floor and the copy-frame baseline, which scores frame i itself against frame $i{+}g$, and Figure 7 visualizes this measurement. The chart wins over copying from $|a|/r_1^* \approx 0.2$, peaks at 1.02, and reaches parity at 1.89. The decoder floor is flat at 0.016 throughout, two to ten times below the signal. The margin is non-monotone in the step, such that at very small steps copying is nearly right, so the advantage is

$\lvert a\rvert/r_1^*$	0.22	0.43	0.63	0.83	1.02	1.26	1.70	2.24
jet (LPIPS)	0.039	0.063	0.080	0.084	0.101	0.130	0.154	0.192
copy	0.047	0.093	0.130	0.115	0.154	0.160	0.173	0.154
decoder floor	0.016	0.017	0.016	0.015	0.017	0.017	0.017	0.015
margin	+0.008	+0.030	+0.051	+0.031	**+0.053**	+0.030	+0.020	−0.038
relative latent error								
full jet	0.937	0.885	0.908	0.917	0.932	0.984	1.162	1.767
first order only	0.990	0.957	0.968	1.006	1.039	1.107	1.311	1.406
ratio	1.06	1.08	1.07	1.10	1.12	1.13	**1.13**	*0.80*

Table 10: IRASim-basis chart on real Bridge frames, binned by $\lvert a\rvert/r_1^*$, median over 1152 transitions. Margin is copy minus jet, where positive means the chart beats predicting no motion. The margin peaks at the radius and reaches zero at $1.89\, r_1^*$. The lower block is an independent confirmation in latent space only.

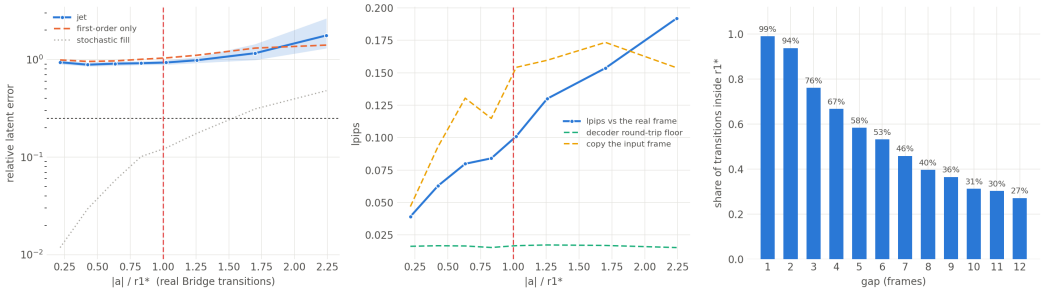

Figure 7: The radius of Action Jet (geometry encoder of which was trained on IRASim generator) measured against real frames with no generator in the loop. *Left:* relative latent error of the full jet against its first-order truncation, with the stochastic fill level. *Center:* LPIPS against the real frame, with the decoder round-trip floor and the copy baseline. *Right:* share of real transitions inside r_1^* by gap. Dashed vertical line is r_1^*.

small, rises to a peak near the radius, and then falls to zero. The empirical radius is the upper end of the interval on which the jet wins, taken as the last positive margin interpolated to zero.

The last block of Table 10 shows that the ratio of the first-order-only error to the full-jet error grows monotonically from 1.06 to 1.13 as the step approaches the radius, which means the correction does more work the further out you go, as expected, and then *inverts* to 0.80 in the last bin. Past r_1^* the cubic term is not merely uninformative, but extrapolating it is worse than truncating the expansion. Figure 8 shows what horizon 4 predictions look like compared to IRASim and WorldGym baselines. We can see that IRASim sometimes moves incorrectly or hallucinates, while WorldGym under-commands and is almost static. Additional qualitative results can be found in Figure 14.

B.4.3 The radius is a property of the chart, not of the generator

We repeat the whole validity radius measurement on the second Bridge chart learned from WorldGym probes. The two charts, one learned from IRASim and the other from WorldGym, are about as unlike as two charts of one dataset can be, as they have nearly orthogonal tangent fields (App. B.5.3, mean per-axis cosine 0.04), translation gains a factor of ten apart, stochastic fill differing by an order of magnitude. And the WorldGym one is the chart that fails to anchor on real transitions at all. If r_1^* were an artifact of one generator's idiosyncrasies, the two would not break down at comparable multiples of their own radii, but they do. Both charts cross the copy baseline between 1.9 and 2.0 times their own r_1^*, both peak at +0.053 LPIPS of advantage, at 1.02 and 1.18, and both show the corrector's advantage growing to about $1.1\times$ and then inverting in the last bin. Both charts give a positive per-state rank correlation, +0.61 and +0.55, and their empirical-to-predicted ratios, 1.72 and 2.02, agree to within 18%. Table 6 in the main paper collects these, Fig. 9 draws both margin curves on the shared normalized abscissa, and Fig. 10 is the WorldGym run in the same three panels as Fig. 7, including the right-hand panel that shows the share of real transitions inside r_1^* at each gap, which is the coverage claim of §4.3 drawn per chart.

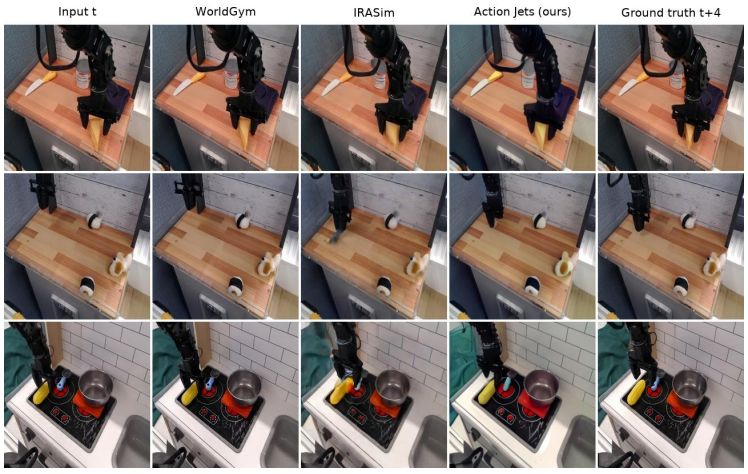

Figure 8: Qualitative results and comparison between WorldGym, IRASim and Action Jet. IRASim often hallucinates or performs slightly wrong motion, while WorldGym under-commands and can show wrong movement.

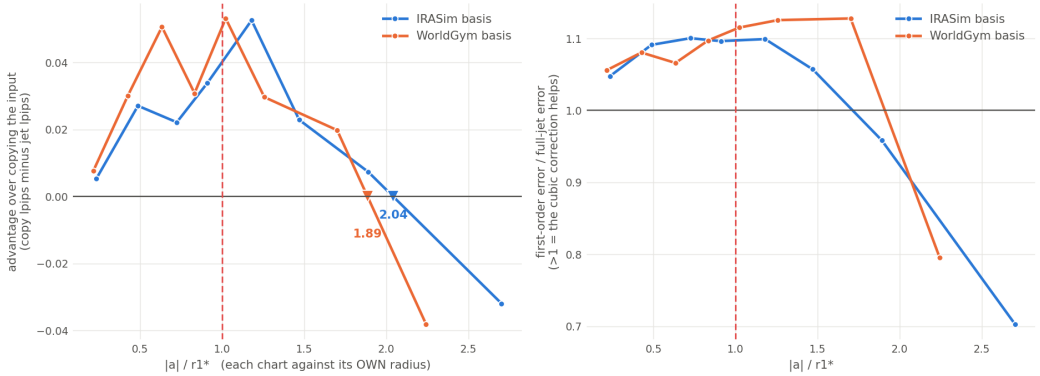

Figure 9: Advantage over the copy baseline for two Bridge charts built from different generators, each against its own r_1^*. The charts have nearly orthogonal tangent fields and translation gains differing by $10\times$, yet both stop beating the baseline at about twice their own predicted radius.

B.4.4 Generator probes on DL3DV

In BridgeData V2 the whole arm is not always visible, and one of the generators (WorldGym) is a negative control. Hence we use DL3DV dataset of camera motion in static indoor and outdoor scenes to measure generator agreement and learned chart consistency. The camera translations and rotations are very different in terms of the visual change that they induce per unit change in action, so to conduct our measurements we instead step each axis to a fraction of its own r_1^*, rendering six intervals at $\{0.25 \dots 1.5\}\, r_1^*$, and in physical units the rotation intervals run $6.9°$ to $41.2°$ and the translations 0.14 to 1.82 chart units. DL3DV translation probe changes the image less than the rotation does, even at $1.5\, r_1^*$, while rotation probes move it five to eight times above the same floor. This is a property of the benchmark since DL3DV scenes are distant enough that camera translation produces little parallax, and it is why the Bridge measurement is the primary one for the validity radius. We report DL3DV degradation on the rotation axes only (Fig. 11). On the rotation axes the Action Jet stays below the induced motion at every rung, beating the copy-frame control, at a ratio of predicted error to induced motion of 0.74–0.85 that is stable across a $6\times$ range of step size. The cubic correction reduces the latent error by 1.9–$3.8\times$ there, far more than on Bridge, because the rotation response is much more nonlinear.

Fitting Eq. 7 to the generator's rendered probe responses and evaluating Eq. 8 on that fit gives a radius that involves no threshold and no chart, a genuinely independent estimate of the same quantity. Across all six axes the generator-implied radius is 0.48 ± 0.03 of the charts's (Table 12),

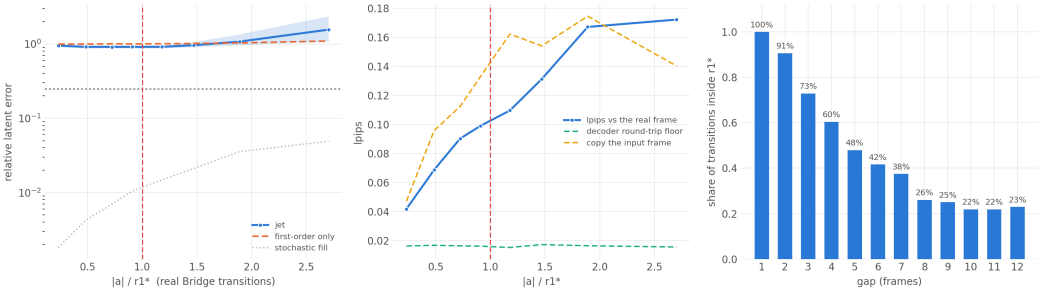

Figure 10: The WorldGym-basis chart, same three panels as Fig. 7: latent error of the full jet against its first-order truncation, LPIPS against the real frame with the decoder floor and the copy baseline, and coverage inside r_1^* by gap. This is the negative control of App. B.5.3, a chart that anti-selects downstream and does not anchor on real transitions, and its radius is nonetheless calibrated. Being well calibrated about your own limits and charting the right object are separate properties, which is why we measure both.

probe axes	probe motion (LPIPS)	decoder floor	verdict
translations t_x, t_y, t_z	0.015–0.109	0.12–0.15	below the floor
rotations r_x, r_y, r_z	0.22–0.85	0.12–0.15	5–8× above
Bridge, for contrast	floor 0.016; every axis moves the image		measurable

Table 11: Did the DL3DV probe move the picture? LPIPS between each probe render and the generator's own render at the base action, against the decoder round-trip floor on the same frames. A measurement below its own instrument's noise floor carries no information, which is what rules the translation axes out on this benchmark. Range over the six ladder intervals, $\{0.25 \dots 1.5\}\, r_1^*$, $n = 64$ states.

a strikingly constant factor. The per-state rank correlation is positive on the translation axes ($+0.24$ to $+0.40$) and absent on the rotation axes (-0.27 to -0.02). This is consistent with the rotation fits being taken across a range where the response is already strongly nonlinear, since almost all rotation in DL3DV produce strong parallax and disocclusions. We report the DL3DV calibration as a scale result and rest the per-state claim on Bridge, where it is measured against real frames. The last column of Table 12 shows that on the DL3DV rotation axes r_1^* sits 7–27× beyond the largest rotation the corrector saw in training, so those radii are extrapolations of Eq. 8 and the ladder exists precisely to put rendered evidence underneath that range, which is done by the 1.25 and $1.5\, r_1^*$ intervals. On Bridge no extrapolation is involved because the ramp reaches $3.1\times$ the radius at p_{95} with a real frame at every interval.

axis	r_1^* (chart)	generator-implied	ratio	ρ per state	$r_1^*/p_{95}(\lvert a\rvert)$
t_x	0.900	0.446	0.50	$+0.40$	0.7
t_y	0.646	0.283	0.44	$+0.24$	1.5
t_z	1.403	0.617	0.44	$+0.28$	1.7
r_x	0.480	0.250	0.52	-0.02	18.4
r_y	0.446	0.220	0.49	-0.27	6.8
r_z	0.426	0.217	0.51	-0.20	26.7
mean			**0.48**		

Table 12: DL3DV: the learned chart's r_1^* against the radius implied by the generator's own probe responses, per axis, $n = 64$ states. The last column gives r_1^* as a multiple of the 95th percentile of the recorded action magnitude.

Table 20 gives r_1^* and the stochastic fill floor per axis for Action Jets, in normalized chart units. The two DL3DV chains, charted from unrelated generators, agree to within about 10% on every axis, showing the radius consequence of the induced-metric agreement in App. B.5.2. And the radii describe the data, where scene-camera rotations in DL3DV produce parallax and disocclusion and leave the linear regime within half a probe step (0.42–0.48), while end-effector rotations in BridgeData V2 move a gripper across a static background and stay linear for three to six (3.0–5.6).

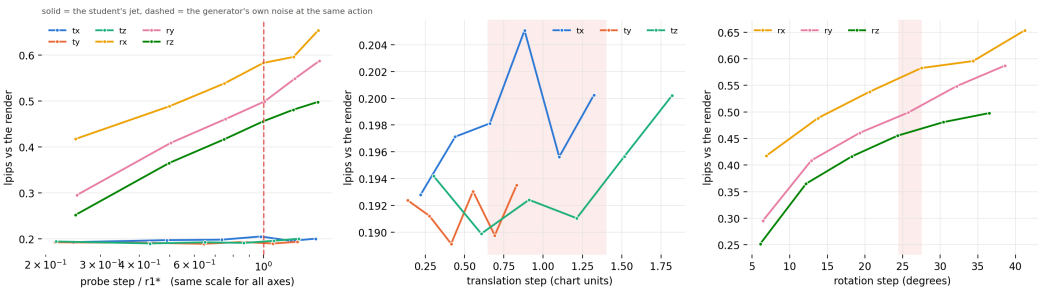

Figure 11: DL3DV degradation across the probe ladder. The chart-to-generator gap is what the ratio of 0.74–0.85 quoted below measures, and it is stable across the whole ladder. *Left:* all six axes on the shared normalized abscissa, log scale, with r_1^* marked, with the flat translation lines measured in Table 11. *Center and right:* the same data against the physical step, split because 0.5 chart units and 0.5° are not the same picture.

The fill floor tracks disocclusion the same way, 3.5–8.7 per unit radius on camera rotations against at most 0.97 anywhere on the robot motion.

B.5 Learned chart analysis

B.5.1 Generators' agreement noise

Two generators cannot agree beyond their own reproducibility, hence we measure it directly by rendering the probes twice with different seeds and comparing each generator with itself. Rendering the LagerNVS ladder a second time at a different seed reproduces the first to seven significant figures, per-axis $\cos(\tau^{(0)}, \tau^{(1)}) = 0.99999952$–$0.99999992$, target norms agreeing to 10^{-6}, because the LagerNVS backbone is feed-forward and deterministic. DepthSplat is deterministic by construction. Both DL3DV generators have reliability $\rho = 1.000$, so the observed generator–generator cosine is the true signal agreement with no attenuation, and the Action Jet' agreement cannot be inflated by noise canceling. This is a stronger position than a measured noise floor would have given us.

On Bridge, WorldGym is stochastic and we measure its reliability from a seed replicate, where $\rho_{\mathrm{WG}} = 0.79$–$0.89$ per axis. IRASim is a video diffusion model too slow to re-render at the required scale, so we bound it instead, and the bound is what lets the Bridge negative stand without a replicate. We model a measured target as signal plus independent noise, and a learned chart that recovers the signal exactly scores $\cos(F, \tau) = \sqrt{\rho}$, and any worse learned chart scores less, so

$$\rho \;\geq\; \cos(F, \tau)^2, \tag{9}$$

a lower bound on a generator's reproducibility computed from the learned charts's fidelity. From the measured $\cos(F_{\mathrm{IRASim}}, \tau_{\mathrm{IRASim}})$ this gives $\rho_{\mathrm{IRASim}} \geq 0.08$–$0.51$ per axis. The bound is loose by construction, so it can only make the ceiling in App. B.5.3 smaller and therefore the negative result harder to claim.

B.5.2 Chart agreement on DL3DV

Table 8 in the main paper gives the medians and Table 13 adds the interquartile ranges. Against a state-permutation null at 0.00 ± 0.015 on every axis, two charts trained on DepthSplat and LagerNVS probes agree at 0.77–0.84 per axis, mean 0.80, while their deterministic generators agree at mean 0.75. On four of six axes the learned charts agree with each other more than their generators DepthSplat and LagerNVS do.

Principal cosines between the two six-dimensional tangent spaces are $(0.87, 0.84, 0.81, 0.75, 0.68, 0.49)$, mean 0.74, so the subspaces largely coincide. The induced metrics $M_{ij} = \langle F_i, F_j \rangle$ agree to a relative Frobenius difference of 0.089 (IQR $[0.066, 0.115]$) with identical eigenvalue ordering, and both charts are similarly anisotropic (condition number 29.1 against 32.0). Since M sets the first-order coefficient in every direction, this sits directly upstream of App. B.4, showing that the two charts do not merely point the same way, but also assign the

axis	Action Jets $\cos(F_A, F_B)$	generators $\cos(\tau_A, \tau_B)$	null
t_x	0.77 [0.71, 0.80]	0.72 [0.62, 0.79]	0.00
t_y	0.81 [0.74, 0.85]	0.73 [0.63, 0.79]	−0.00
t_z	0.77 [0.70, 0.81]	0.68 [0.56, 0.75]	−0.00
r_x	0.84 [0.78, 0.87]	0.69 [0.64, 0.73]	−0.00
r_y	0.82 [0.79, 0.85]	0.84 [0.80, 0.87]	0.00
r_z	0.79 [0.72, 0.84]	0.86 [0.82, 0.89]	−0.00
mean	**0.80**	0.75	0.00

Table 13: Cross-generator tangent agreement on DL3DV, median [IQR] over 256 states, probed at $0.25\, r_1^*$ per axis. Action Jets column compares the two Action Jet charts learned from DepthSplat and LagerNVS probes. Generators column compares the DepthSplat and LagerNVS generators that supervised them, both of which are deterministic (App. B.5.1).

same lengths. It is also why r_1^* is evaluated along the traveled direction rather than combined from per-axis radii, since with a condition number near 30, an isotropic summary of the radius would be wrong by a factor of five.

Two charts supervised by different generators being closer to each other than the generators are is consistent with each encoder discarding the component of its supervision that the other generator does not share, since the generator-specific part of a measured field is, from the encoder's point of view, unpredictable structure, and a fit under the loss of Eq. 3 has no reason to preserve it. This is not explained by shared architecture alone, because the two chains use the same encoder architecture on Bridge as well, and those charts do not agree (App. B.5.3), making Bridge's negative result function as the control for this positive one. The measurement above steps each axis to 0.25 of its own radius. The same comparison at a 1σ probe on a band-square window and 148 states (Table 14) gives generators 0.84 and Action Jets 0.82, arriving at the same conclusion, with the Action Jets' margin over the generators reversed in sign and within the spread. The two measurements are not interchangeable and we do not average them, since per App. B.4.4, 1σ is a different physical step on every axis and sits past breakdown on the translations. We report the r_1^*-normalized ladder in the main paper because it is the only scale at which all six axes are comparably far from their own breakdown, and quote the σ-probe here as the robustness check.

pair	t_x	t_y	t_z	r_x	r_y	r_z	mean	n
generator ↔ generator								
DL3DV: DepthSplat ↔ LagerNVS	.81	.82	.81	.88	.89	.85	**.84**	148
Bridge: IRASim ↔ WorldGym (span)	–	–	–	–	–	–	**.036**	148
generator ↔ probed baseline								
DL3DV: LagerNVS ↔ FrameCrafter	.67	.60	.67	.78	.79	.69	.70	148
DL3DV: DepthSplat ↔ FrameCrafter	.59	.55	.60	.72	.75	.63	.64	148
Action Jet ↔ Action Jet								
DL3DV pair, charted separately	.79	.83	.78	.86	.84	.82	**.82**	148
Bridge pair, charted separately	−.02	.05	.07	−.10	.09	.14	**.039**	148

Table 14: Robustness check for charts agreement: the same comparisons at a 1σ probe, band-square window on scenes, 148 states. FrameCrafter is probed but not charted. Conclusions are unchanged, the absolute values are not comparable with Table 13, because the two ladders step the axes by different physical amounts.

B.5.3 Chart disagreement on BridgeDataV2

Listed are the measurements §4.3 refers to regarding the generator and Action Jet chart disagreement on BridgeData V2. We learn a chart from IRASim and another from WorldGym, then probing both on 148 identical held-out Bridge states, map each generator's action convention into a common frame, and compare the tangent fields exactly as on DL3DV. The two Action Jet charts agree at -0.02 to 0.12 per axis at the permutation null, and so do their *generators*, at -0.02 to 0.05. As established by Eq. 9, this result is not a noise artifact. Two generators measuring a common signal would agree at $\sqrt{\rho_A \rho_B} \cdot \cos(s_A, s_B)$. With ρ_{WG} measured and ρ_{IRASim} bounded below, the ceiling is at least 0.25–0.67 depending on axis, against an observed 0.02–0.05 (Table 15). The noise-

free tangent responses of the two generators therefore have cosine at most 0.04–0.21, so IRASim and WorldGym genuinely define different tangent structures on Bridge. Each Action Jet chart is meanwhile faithful to its own generator, with $\cos(F, \tau) = 0.28$–0.71 for IRASim and 0.49–0.65 for WorldGym, the latter against a measured ceiling of $\sqrt{0.79} = 0.89$, so the disagreement is inherited rather than introduced.

axis	Action Jets	generators	$\rho_{\mathrm{IRASim}} \geq$	ρ_{WG}	ceiling $\geq$	$\cos(s_A, s_B) \leq$
t_x	-0.02	-0.02	0.48	0.87	0.65	0.04
t_y	0.06	0.03	0.51	0.87	0.67	0.05
t_z	0.08	0.05	0.40	0.89	0.59	0.08
r_x	-0.09	-0.04	0.19	0.81	0.39	0.11
r_y	0.09	0.04	0.23	0.84	0.44	0.10
r_z	0.12	0.05	0.08	0.79	0.25	0.21

Table 15: The two generators for BridgeData V2 do not share a tangent signal. ρ_{WG} is measured from a seed replicate, while ρ_{IRASim} is the lower bound $\cos(F, \tau)^2$ of Eq. 9, since IRASim is a diffusion model too slow to re-render. The final column is the implied upper bound on the cosine between the two generators' noise-free responses.

Because the two generators express actions in different conventions, a per-axis cosine could in principle understate their agreement, so a consistent remixing of the six axes would destroy it while leaving the underlying surface identical. We therefore also compare the subspaces their six fields span, with a principal-angle overlap invariant to any such remixing (1 for identical subspaces). It is 0.036, against 0.573 when one generator is probed twice and 0.001 for random fields, and the disagreement passes unchanged to the Action Jets (0.039). A per-state 6×6 mixing fit gives the same result (0.12 recovered, against 0.78 on the DL3DV pair).

Table 16 shows that the Action Jet chart extracted from IRASim is the one that belongs to the data. Fit to the generator cannot tell the two charts apart (0.513 and 0.542), so any distillation-style metric would call them equally good. Anchoring against real held-out transitions separates them immediately, showing that chart learned from IRASim scores cosine $+0.249$ against a rolled-action control of -0.014, a gap of $+0.263$, with least-squares gain 0.74, while the one learned from WorldGym scores -0.011 against a control of $+0.088$, a gap of -0.099, with gain -0.10, showing it is a chart whose best linear fit to the data points backwards. Re-probing raised the WorldGym Action Jet's fit to its own generator from 0.39 to 0.54 and did not move its anchoring, so fitting the generator better did not bring it closer to the data. We keep this result as a negative control for everything downstream, with §4.3 using it as a verifier.

	IRASim Action Jet	WorldGym Action Jet
held-out fit to own generator's fields	0.513	0.542
real-transition anchoring (cos, mid-band, n=48)	$\mathbf{+0.249}$	$\mathbf{-0.011}$
rolled-action control	-0.014	$+0.088$
anchoring gap (true − control)	$\mathbf{+0.263}$	$\mathbf{-0.099}$
least-squares gain α^* (median)	0.74	-0.10
tangent vs. copy floor (PSNR, dB)	13.49 vs. 12.94	12.84 vs. 12.94

Table 16: Fit to the generator cannot tell the two Action Jets apart, so a distillation metric would call them equally good, but anchoring against real held-out transitions separates them at once. The rolled-action control pairs each state's fields with another state's action, so it measures the anchoring any two arrays would show. The least-squares gain α^* is the scalar that best maps the chart's predicted displacement onto the real one, where negative means the chart's best linear fit to the data points backwards.

B.6 Extended DL3DV results

Table 21 is the full per-gap grid behind Table 7, with SSIM, tail LPIPS over the hardest third of states, and paired bootstrap intervals against our LagerNVS-probed learned chart. Table 22 ablates the chart itself at all four gaps. The charts learned from DepthSplat and LagerNVS differ by at most 0.006 LPIPS and 0.09 dB at every gap, which is consistent with the agreement result of App. B.5.2, showing that two charts that share no generator make the same predictions.

The rows showing results for autoregressively composed prediction lose 0.042 to 0.111 LPIPS against a single extrapolated query on every interval, because re-encoding each decoded frame compounds error faster than the calibrated jet loses accuracy in extrapolation. This is from the Bridge finding in §4.3, and App. B.4.4 explains the split, where on Bridge a four-step chunk exceeds the translation radii while each single step stays inside them, so autoregressive composing is viable, while on DL3DV scenes even one step may exceed the rotation radii of 0.42–0.48, so there is no inside to stay in and the extra encode is pure loss.

Table 1 scores geometry rather than pixels, and we describe the full evaluation protocol here. Each predicted frame is paired with the real frame at the same target pose and the pair is matched with MASt3R. Then PnP on those matches recovers the relative camera between prediction and ground truth frame, whose ground truth error is the identity, and we report the median rotation error and the virtual-correspondence reprojection error (VCRE) of that camera. A frame matched against itself gives the instrument's own noise at 0.12° and 0.4 px, below which no method can be distinguished. And the copy control scores the input frame as if it were the prediction, which is exactly the camera motion a predict-nothing model leaves uncorrected, without it, a low rotation error would be unreadable, since a method that changes nothing on a slow camera path would score well. All images use the common 256-pixel window. Figures 12 and 13 show the predictions these numbers summarize, with the MASt3R correspondences that Table 1 scores drawn on top.

B.7 Instrument ladders, controls, and encoder fidelity

Table 17 probes DepthSplat at six displacements from 0.25σ to 6σ and compares per-unit fields at adjacent intervals. Over this $24\times$ range the surface never reads as flat: the mean adjacent-rung cosine is $0.69, 0.59, 0.41, 0.50, 0.29$, and the most curved axis is pan, the one with the largest image motion per unit action. Pan is also the one axis whose cosine is already low at the smallest interval (0.33 at $0.25 \leftrightarrow 0.5$ against 0.67–0.83 for the others), so on that axis there is no interval in this ladder that behaves linearly at all. The one non-monotone entry $2 \leftrightarrow 3$ reads higher than $1 \leftrightarrow 2$ because the intervals are not uniformly spaced. An Action Jet trained at $f{=}1$ scores 0.82 against held-out fields probed at 1σ and falls away in both directions, to 0.58 one octave below and 0.43 one octave above, reaching 0.11 at 6σ. A chart extracted at one displacement is a chart of that displacement. Hence everywhere we probe at the per-step action scale used at deployment and never compare generators probed at different displacements, because two different generators probed at the same δ agree more (0.84, Table 14) than one generator agrees with itself across neighboring scales (0.41).

rung pair	t_x	t_y	t_z	r_x	r_y	r_z	mean
$0.25 \leftrightarrow 0.5$	.67	.78	.73	.83	.33	.83	.69
$0.5 \leftrightarrow 1$	.52	.67	.62	.78	.22	.74	.59
$1 \leftrightarrow 2$	.36	.53	.49	.37	.25	.49	.41
$2 \leftrightarrow 3$	.44	.59	.57	.46	.42	.54	.50
$3 \leftrightarrow 6$	.26	.31	.34	.23	.40	.24	.29
Action Jet trained at $f{=}1$, held-out cosine at probe scale $0.25/0.5/1/2/3/6$							
	.43	.58	**.82**	.43	.23	.11	

Table 17: Full curvature ladder (DepthSplat, DL3DV, 148 states): per-axis cosine between per-unit fields at consecutive probe magnitudes, and an Action Jet trained at $f{=}1$ evaluated against every interval. Curvature tracks image flow per unit action, while pan, the largest-flow axis, is most curved, and the Action Jet peaks sharply at the displacement it was trained on and collapses off-scale.

Table 18 lists each control and the artifact it would expose. The random-field floor (0.001) prices dimensionality and the shared autoencoder, where two unrelated fields in this latent space score zero, so a cosine of 0.80 is not a high-dimensional artifact. The seed-paired repeat (0.573 on Bridge, 1.000 on DL3DV) is the ceiling every agreement is read against, and Eq. 9 supplies it where a repeat is unaffordable. The per-state $6{\times}6$ mixing fit (0.12 against 0.78 on the DL3DV pair) and the kinematic convention certification ($+1.0000$ against -0.72 for the identity map) together close the obvious escape route from the Bridge negative, where that the two generators agree but express it in different coordinates. The copy-frame control (12.94 dB) is the baseline the validity radius is defined against. And the oracle-of-K ceiling (0.006) prices breadth itself, so the selection results of §4.3 report how much of an available gain each verifier captures rather than an unanchored MSE.

control	artifact it would expose	measured
random-field floor	cosine inflated by dimensionality or the shared VAE	0.001
seed-paired repeat, same generator	disagreement that is only sampling noise	0.573
split-half reliability ceiling $\sqrt{r}$	a stochastic generator penalized for its noise	0.86
per-state 6×6 mixing fit	disagreement recoverable by a linear relabelling	0.12 vs. 0.78
kinematic convention certification	generators compared in incompatible action frames	$+1.0000$ vs. -0.72
rolled-state null	an Action Jet emitting a generic, state-independent field	0.00–0.09
rolled-action control	anchoring that any two arrays would show	-0.014
copy-frame control	a prediction no better than repeating the input	12.94 dB
oracle-of-K ceiling	selection gains that breadth alone would deliver	0.006
certified null arm	an effect that survives a chart known to be wrong	-0.011
reconstruction bound	treating re-rendering methods as competitors	upper bound

Table 18: Controls, and the artifact each one rules out. Every agreement measurement in the paper is reported against a floor and, where the generator is stochastic, a ceiling.

Table 19 reports each geometry encoder's held-out field cosine against its own generator, with a rolled-state null beside it. The nulls sit at 0.00–0.09, so none of these encoders is emitting a generic state-independent field. The DL3DV charts roughly double the Bridge charts, and the reason is the generators rather than the encoders. On DL3DV both generators are deterministic (App. B.5.1), so their probe targets carry no sampling noise and a perfect encoder would score 1.0. On Bridge WorldGym is stochastic with measured reliability $\rho_{\mathrm{WG}} = 0.79$–$0.89$, which caps any chart at $\sqrt{0.79} = 0.89$, so its learned chart reaches 0.54, or about 61% of a ceiling. The same argument cannot be run for IRASim without circularity, since its reliability is bounded from this quantity (Eq. 9).

The DepthSplat chart scores 0.82 on the full window and 0.73 on the square one, the same chart and the same generator, differing only in where it is scored, so the only clean cross-generator row pair here is DepthSplat-square against LagerNVS-square, 0.73 against 0.77. This is the same window the cross-generator protocol of §3.1 imposes, and the 0.09 swing is a measure of what ignoring it would cost. Encoder latency is 8.3–15.9 ms per state, which is the dominant term in the 37.9 ms single-candidate cost of Table 5, everything after it is arithmetic.

Action Jet	window	t_x	t_y	t_z	r_x	r_y	r_z	mean	null
DL3DV / DepthSplat	full	.80	.82	.79	.86	.83	.84	**.82**	.01
DL3DV / LagerNVS	square	.73	.78	.73	.82	.79	.78	**.77**	.00
DL3DV / DepthSplat	square	.69	.73	.69	.78	.75	.74	**.73**	.00
Bridge / IRASim	–	.65	.68	.60	.41	.44	.30	**.51**	.04
Bridge / WorldGym	–	.55	.56	.62	.51	.54	.47	**.54**	.09

Table 19: Geometry-encoder held-out field cosine against the Action Jet's own generator, with rolled-state nulls at right. DL3DV Action Jets roughly double Bridge Action Jets, tracking generator determinism and conditioning tightness. Encoder latency is 8.3–15.9 ms per state.

C WHY THE JACOBIAN HAS TO BE MEASURED

We tested the natural alternative to a dedicated encoder, namely read-out heads on the frozen generator's features, at several depths and capacities. All of them fail, and the failure is diagnostic rather than an optimization artifact: the map from generator features to the per-block generator field is ill-posed, in that blocks carrying near-identical generator features require nearly unrelated generators. Nothing that reads those features pointwise can separate them. Permitting gradient to flow into the generator does reshape its features, at a measurable cost to its predictive accuracy, and still does not supply the missing structure. A model trained to predict action-conditioned frames is under no obligation to represent $\partial\,\mathrm{output}/\partial\,\mathrm{action}$ explicitly, and it does not. This is the precise sense in which our method is not distillation, because there is no representation in the generator to copy, so the Jacobian is obtained by measurement.

Why probe a black box rather than differentiate it. On a differentiable generator one could read exact Jacobians by automatic differentiation, and as an ablation we verify that the symmetric probe agrees with autodiff where both are available (§4.2), but we do not require it. The probe treats the

generator as a black box, so any model mapping (state, action) to a prediction can serve, including API-only models and multi-step samplers, and the finite difference is a mollified derivative that averages generator noise over a small but finite δ rather than differentiating it at a point.

Dispensing with the generator and regressing $z' - z$ on observed action deltas founders on identifiability, where real deltas are correlated and multi-factor, so per-factor fields entangle exactly where factors co-occur, which is where the geometry is most interesting. The generator answers isolated single-axis queries at arbitrary states, queries the dataset never contains, and that isolation is what makes the recovered $\widehat{J}_i$ a per-factor derivative rather than a regression coefficient on correlated inputs.

Parity separation (§3.3) is identifiable only if, conditioned on the state, training deltas occur with both signs. Under history conditioning this fails silently, where the context pair reveals the ongoing motion and every training pair continues it, so the per-state delta distribution is one-sided even when signs balance marginally across clips. Fitted odd and even coefficients then entangle, correct along the observed continuation and arbitrary along the reversal, and query quality becomes direction-asymmetric, which we observed directly as growing artifacts when sliding against a clip's own motion. Plain time reversal does not fix it, because reversing flips context and delta together and leaves counter-momentum queries untrained.

D METRIC VALIDITY FOR POLICY CONSISTENCY

Write π for a frozen policy and f_{t+h}, $\hat{f}_{t+h}$ for real and predicted futures, and define $\mathrm{ADE} = \frac{1}{H}\sum_h \|\pi(\hat{f}_{t+h}) - \pi(f_{t+h})\|_2$ under paired sampling keys, so the policy's own sampling noise cancels from the comparison. The obvious alternative, comparing the policy's action on the predicted future against the demonstrated action, may seem equivalent at the first glance but it is not. If we consider what a perfect simulator scores, on the metric we use it scores zero, because predicted and real futures are then identical, but on the demonstrated-action metric it scores the policy's own cloning error and does not optimize the metric at all. This is evidenced by the fact that our copy-frame control beats the real frames on it (1.93 against 2.13) by exploiting the temporal smoothness of demonstrations. A metric whose optimum is not at the true simulator cannot rank world models, so we report it as context only.

	DL3DV (Lager chain)		DL3DV (DepthSplat chain)		Bridge (IRASim chain)
axis	r_1^*	fill	r_1^*	fill	r_1^* / fill
t_x	0.80 [0.68,0.92]	1.2	0.88 [0.73,1.07]	1.4	1.89 [1.65,2.14] / 0.71
t_y	0.53 [0.45,0.64]	1.3	0.55 [0.50,0.68]	1.3	2.10 [1.83,2.44] / 0.30
t_z	1.16 [1.02,1.34]	1.1	1.21 [1.08,1.54]	1.4	2.13 [1.79,2.39] / 0.97
r_x	0.47 [0.45,0.50]	8.7	0.48 [0.44,0.52]	8.5	3.33 [2.80,4.04] / 0.18
r_y	0.44 [0.42,0.46]	3.5	0.45 [0.41,0.47]	3.7	3.02 [2.44,3.68] / 0.10
r_z	0.43 [0.41,0.45]	6.3	0.42 [0.41,0.44]	8.1	5.63 [5.00,6.36] / 0.04

Table 20: Self-reported validity radii r_1^* (median [IQR]) and stochastic fill floors per unit radius, in normalized chart units where 1.0 is one probe displacement. The normalization is what makes the two embodiments comparable.

E GOAL-IMAGE VERIFICATION UNDER A COMPUTE BUDGET

Planning utility is fidelity times the breadth of futures a planner can afford to weigh, and the cost structure lets us measure the trade. For each state a pool of candidate action chunks is scored by each world model as a goal-image verifier, where we predict every outcome, rank by similarity to the true $t{+}4$ frame, and judge the selected candidate against the demonstrated action, a ground truth no learned proxy can bias, with random-of-K and oracle-of-K bracketing every curve (Table 4). Breadth pays for competent verifiers, where every functioning model improves monotonically with pool size and the oracle falls $90\times$ from $K{=}1$ to 128, which prices breadth itself. Verification skill dissociates from simulation fidelity, where WorldGym anti-selects, its choices worsening as the pool grows despite mid-pack pixel metrics, while the chart at $150\times$ less cost per candidate matches IRASim at small pools and passes it at $K{=}32$. Under a fixed budget the ranking inverts, since

method	g	n	PSNR↑	SSIM↑	LPIPS↓	tail LPIPS↓	ΔLPIPS vs. ours [95% CI]
Copy input (control)	1	370	18.63	0.506	.122	.220	$[-0.045, -0.035]$
	2	370	16.02	0.372	.205	.347	$[-0.008, +0.005]$
	3	370	14.83	0.313	.267	.422	$[+0.011, +0.025]$
	4	370	14.11	0.282	.315	.471	$[+0.025, +0.039]$
DepthSplat	1	370	25.64	0.862	.047	.076	$[-0.119, -0.110]$
	2	370	23.17	0.800	.075	.125	$[-0.138, -0.127]$
	3	370	21.13	0.744	.104	.171	$[-0.151, -0.138]$
	4	370	19.55	0.695	.133	.212	$[-0.157, -0.142]$
LagerNVS	1	370	30.45	0.930	.039	.057	$[-0.128, -0.119]$
	2	370	27.38	0.879	.062	.099	$[-0.151, -0.139]$
	3	370	25.24	0.826	.088	.142	$[-0.167, -0.154]$
	4	370	23.65	0.779	.115	.181	$[-0.175, -0.159]$
FrameCrafter	1	370	17.43	0.447	.187	.184	$[+0.029, +0.064]$
	2	370	15.36	0.343	.296	.282	$[+0.108, +0.162]$
	3	370	14.34	0.304	.366	.375	$[+0.142, +0.202]$
	4	370	13.66	0.277	.413	.466	$[+0.157, +0.222]$
Ours (LagerNVS generator)	1	370	20.40	0.600	.162	.217	–
	2	370	18.67	0.507	.207	.283	–
	3	370	17.50	0.438	.249	.338	–
	4	370	16.76	0.390	.283	.377	–
Ours (DepthSplat generator)	1	370	20.48	0.597	.164	.217	$[+0.001, +0.003]$
	2	370	18.69	0.503	.211	.286	$[+0.002, +0.005]$
	3	370	17.53	0.435	.253	.342	$[+0.002, +0.005]$
	4	370	16.78	0.387	.287	.381	$[+0.003, +0.006]$

Table 21: All methods, causal tier (two trailing context frames), common 256-pixel window. The copy control and the difficulty strata use exactly our states. Tail LPIPS averages the hardest third of states, ranked by how much the copy control misses. ΔLPIPS is paired (row minus ours) on shared states; negative means better than ours. FrameCrafter is evaluated on a subset of our states because of its sampling cost. An interpolation tier (six surrounding context views) is not reported: its context includes the target frame at $g \geq 2$.

	PSNR↑				LPIPS↓			
gap g	1	2	3	4	1	2	3	4
LagerNVS generator								
single extraction, FT decoder	21.2	19.7	18.5	17.6	.139	.172	.209	.243
composed (AR), FT decoder	21.2	18.6	16.8	15.5	.139	.217	.284	.344
composed (AR), frozen decoder	21.0	18.0	16.1	14.8	.158	.273	.388	.478
DepthSplat generator								
single extraction, FT decoder	21.4	19.7	18.5	17.6	.142	.177	.214	.249
composed (AR), FT decoder	21.4	18.9	17.1	15.9	.142	.222	.292	.352
composed (AR), frozen decoder	21.1	18.2	16.3	15.0	.158	.281	.410	.505
full window, frozen decoder (DepthSplat generator)	17.7	16.9	16.3	15.6	.213	.285	.350	.407

Table 22: Ablations of our chart on the $n=100$ states per gap on all four gaps. Composing four single-step queries autoregressively is worse than one extrapolated query at $g \geq 2$ (paired ΔLPIPS $+0.042$ to $+0.111$, every CI excluding zero): re-encoding each decoded frame compounds error faster than the calibrated jet loses accuracy in extrapolation. The fine-tuned decoder's gain grows under composition (LPIPS $+0.016$ at $g=1$ to $+0.153$ at $g=4$, frozen minus fine-tuned), because every step re-encodes the previous decode. At $g=1$ both extraction modes coincide exactly, a pipeline-identity check. The full-window row is not decoder-matched and is shown for feasibility only.

one second buys IRASim a fifth of a candidate and the chart its entire pool. On a wider pool of 128 candidates the composed protocol, which stays inside the radius, edges out single extraction at every pool size (0.045 against 0.051 at $K=128$), and averaging over noise draws changes nothing, consistent with the low fill floors on this domain.

	decoder	gap 1		gap 2		gap 3		gap 4	
free-form	frozen	33.54	.0325	30.32	.0638	27.53	.0942	26.24	.1249
free-form	FT	34.45	.0153	31.19	.0267	28.28	.0391	27.26	.0511
free-form+pixFT	frozen	32.45	.0285	28.73	.0558	25.96	.0884	24.54	.1275
free-form+pixFT	FT	33.49	.0165	30.17	.0274	27.72	.0381	26.77	.0484
2-jet+pixFT	frozen	28.69	.0332	26.09	.0680	23.65	.1033	22.76	.1373
2-jet+pixFT	FT	30.46	.0213	27.93	.0350	25.86	.0468	25.33	.0575
3-jet	frozen	28.98	.0319	26.31	.0643	24.07	.0952	23.13	.1272
3-jet	FT	31.80	.0183	29.05	.0305	26.92	.0403	26.27	.0501

Table 23: Kubric full ablation ladder. Corrector ladder $\times$ decoder (PSNR / LPIPS per gap). The shared decoder was fine-tuned on {3-jet, free-form+pixFT} latents; the out-of-mix plain free-form model rendering best through it evidences that the shared renderer plays no favorites.

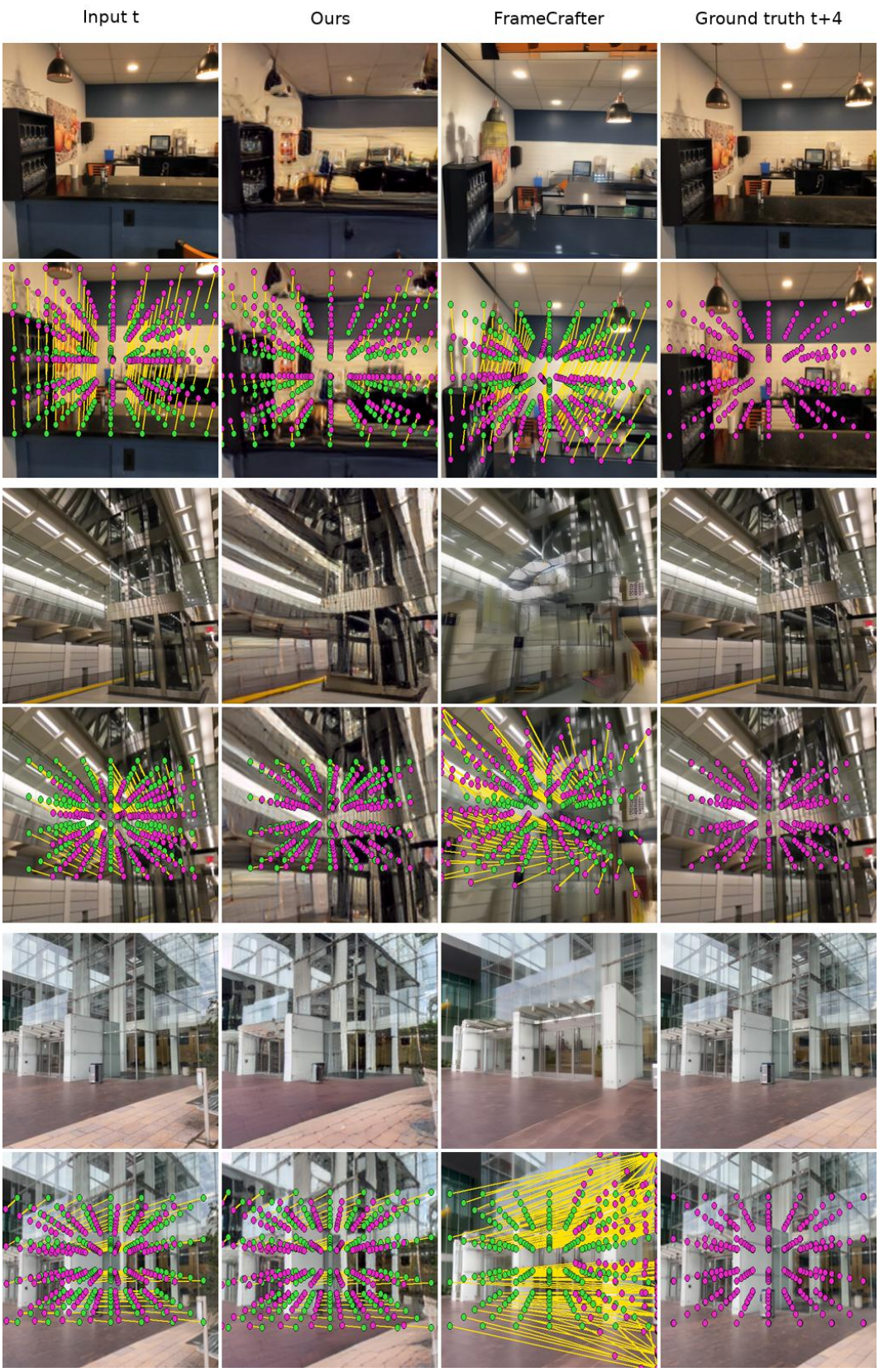

Figure 12: DL3DV forward prediction and geometric consistency qualitative results. Held-out scenes from DL3DV Benchmark dataset at target frame $t + 4$ compared to Action Jet and FrameCrafter predictions. The overlay marks the MASt3R correspondences measured in Table 1. Action Jet may blur high-frequency details or leave slight artifacts at repeated patterns, while the diffusion baseline shows a causal and reprojection error.

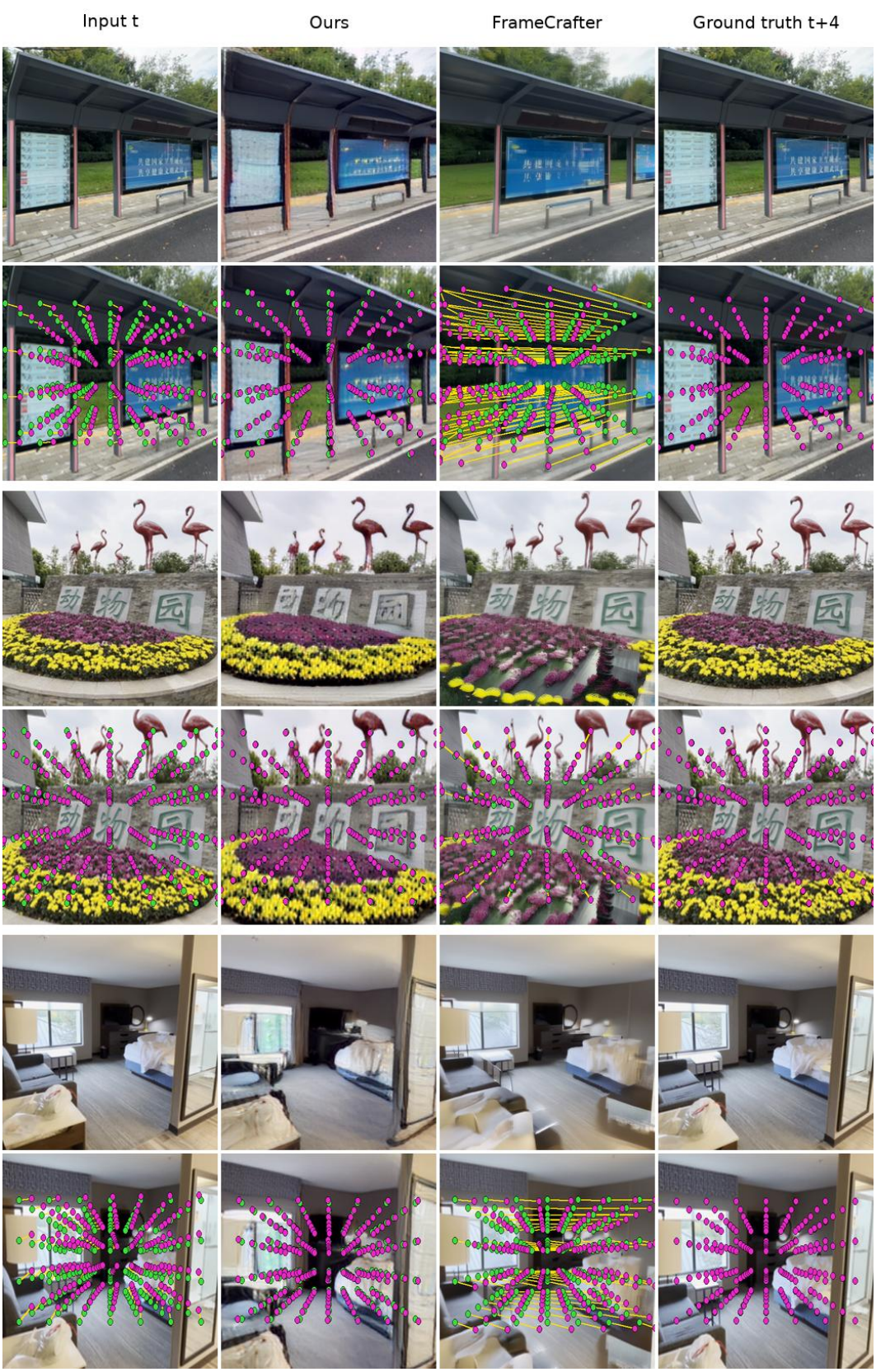

Figure 13: Additional DL3DV forward prediction and geometric consistency qualitative results, continued from Fig. 12.

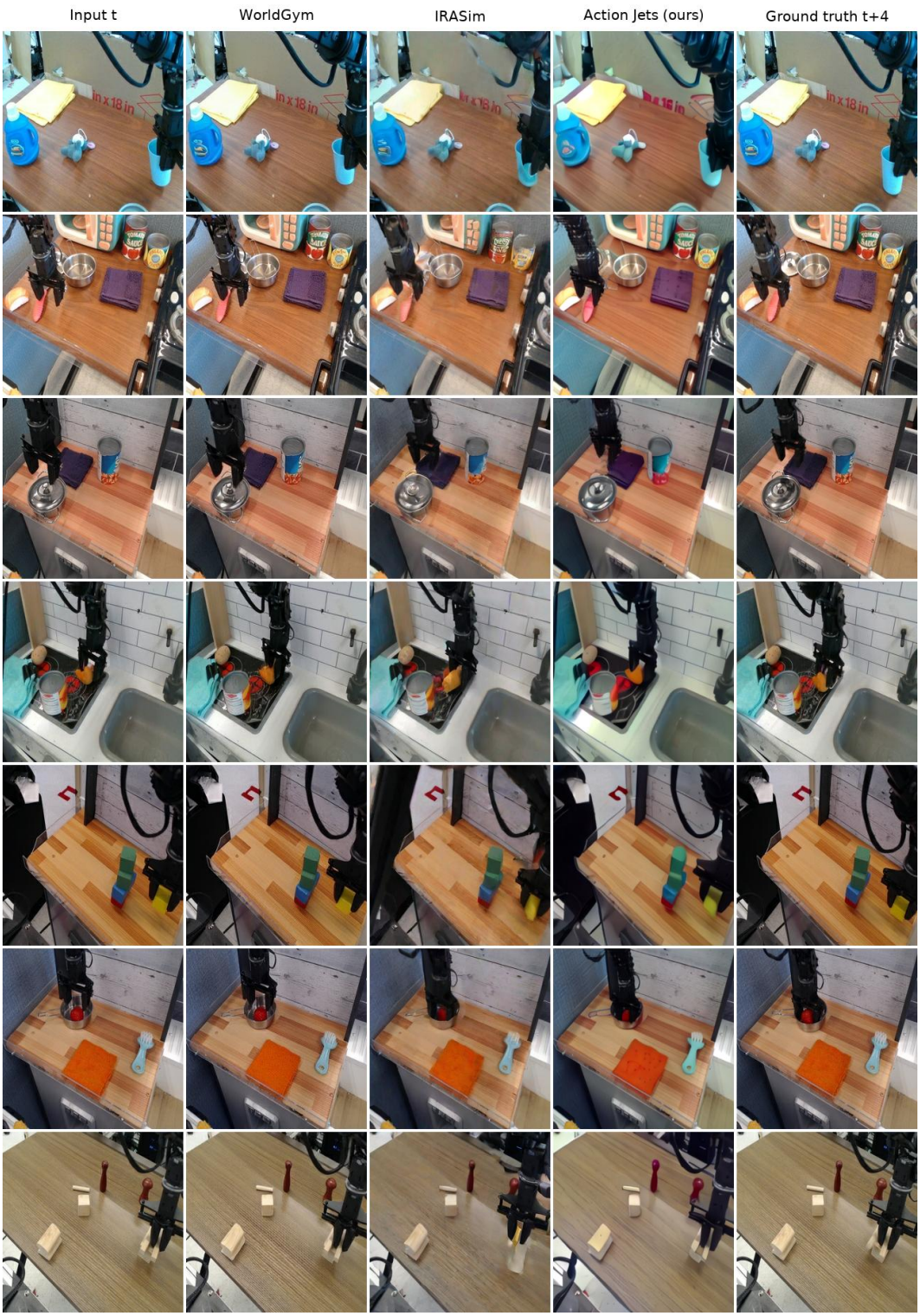

Figure 14: Additional qualitative results for BridgeData V2 open-loop rollouts. Held-out episodes under the recorded actions at gap 4 against the real frames and the source generators. These are the rollouts scored in Table 3.