

Action Forcing: Training World Models on Unsupervised Video by Recovering Underlying Egomotion Bases

Ashish Sundar^{1,2} Tiankuo Hou¹ Zhong Fan¹
Chunbo Luo¹ Xiaoyang Wang¹

¹University of Exeter, Exeter, United Kingdom

²Environmental Intelligence CDT

Abstract

We are living through a golden age of videos: cameras are ubiquitous, storage is inexpensive, and vast amounts of footage are recorded every day. Yet, the synchronised action annotations needed to train controllable world models remain elusive. Existing approaches make use of instrumented platforms with calibrated sensors, costly manual annotation, or latent-action models which lack grounding. We instead turn ordinary unlabelled video into action-supervised training data by recovering (without training) a data-derived egomotion basis. We track pixel displacements across frames and exploit the recurring coherent structure induced by egomotion to obtain grounded control signals directly. Using a method as simple as principal components analysis perform this, we find that the leading components provide signed, scalable, and composable throttle–yaw controls, although the method can recover only motion axes represented in the data. To prevent a high-capacity video DiT from exploiting pixel-level supervision, an online latent critic distils a frozen decoder–tracker–PCA (Principal Components Analysis) teacher without backpropagating through the decoder or tracker. Finally we critique the use of video generation metrics to evaluate WMs and introduce an example of an alternative, reference-free evaluation method. We measure *controllability*, *plausibility*, *conjuring* (creating objects out of thin air) and *geometric integrity*, revealing failures that conventional video metrics miss. We show that most baselines follow familiar action directions but struggle to reverse or remain stationary. Our model handles both while retaining compositional control and generation quality. Despite backwards actions being less than 1% of our training data, we find that the model learns to reverse, scale its response linearly, and compose throttle with steering, all simply by learning through a grounded action space. Cross model action forcing shows that most evaluated systems either fail to reverse and perhaps surprisingly, struggle to stay still. Our model does both consistently. Ultimately, our results show that unlabelled real world video can be used to train WMs, removing the shackles of action labels from WM research and training.

1 Introduction

Large video diffusion transformers have recently found much success in learning powerful visual and temporal priors, but converting them into world models remains challenging without good action supervision. To get supervision, new data either needs to be painstakingly collected [1–6], or training is relegated to the domain of videogames [7–12]: synthesised worlds that can only try to mimic the complexity and

stochasticity of the real world, but are not suitable for real world applications. While some [7, 9, 13, 14] use latent action models to bridge the gap, these often prove to lack any meaningful grounding (cross context action identifiability). Tellingly, even high performing systems retreat from messy data when they can: Hong et al. [15] abandons real world walking and drone data for unreal engine generated data, because real world data is dominated by forward motion and motion entangled across different dimensions.

We train directly on this difficult regime. FrodoBots video [16] contains low-speed rover journeys through real cities, with pedestrians, uneven ground, occlusion, weather, lens contamination, and unstructured interactions. Its recorded commands are significantly affected by network latency and do not reliably describe realized motion; we find that the labels actively destabilise training. We therefore treat the videos as unlabelled and recover control inputs from the motion rendered in the pixels.

Our approach rests on one primary idea - the pixel movements within frame follow the same broad stroke patterns for the same egomotion directions. When principal components of these pixel vectors are taken over a large corpus, larger egomotion patterns emerge clearly. We use this as an action space as it ends up being linearly scalable, invertible, and composable across multiple directions. We use the two highest variance component weights as the egomotion input and thus avoid conditioning on dense flow or raw tracks directly [17–19]. This means we do not need to construct a full target motion field at inference time.

We instantiate the method by adapting Wan2.1 on a coherent 88 hour subset composed of 11 cities from the larger 2000 hour FrodoBots video dataset. Despite finding only approximately 55 minutes (1% of the total footage) of sustained reverse motion in the full corpus, the resulting model learns backward control and combines it well with steering. Representative qualitative rollouts are provided in Appendix A.

Our contributions:

1. We recover an interpretable, signed, and composable egomotion action space directly from unlabelled real-world video using point tracking and principal components analysis (PCA), without recorded actions or a learned latent-action model.
2. We introduce an online latent egomotion critic that distils pixel-space motion into differentiable latent supervision, grounding action following without backpropagating through the video decoder or tracker.

3. We show that standard video evaluation metrics are inadequate for world models, where action forced rollouts have no paired reference to judge against. We introduce an example of a reference-free action-forcing evaluation that measures controllability, geometric integrity, and plausibility.

2 Related Work

World models from pretrained video generators. Recent work adapts pretrained video DiTs into action-conditioned rollout models using camera poses, keyboard–mouse inputs, game controls, or curated robot actions [20–28]. We compare directly against minWM, WorldCam, Astra, Matrix-Game 2.0, HY-WorldPlay, and Yume. These systems demonstrate strong controllability when actions are available, but rely on labelled, instrumented, or simulated control signals. We instead recover a grounded egomotion interface directly from unlabelled real-world video.

Actions recovered from video. Latent-action models learn controls through inverse dynamics without recorded labels [13, 14, 29, 30], but their codes can be opaque and inconsistent across contexts [9]. Motion-conditioned generators instead recover optical flow or dense point tracks from video [17–19], but these representations are high-dimensional and require a full motion field at inference time. We compress tracked motion with PCA into a signed, low-dimensional action space that supports direct commands, reversal, and composition.

VLM-based evaluation of generated video. Recent metrics use multimodal models or learned evaluators to assess generated content [31–35]. We instead use narrow reference-conditioned probes for specific world-model failures and validate each instrument against human annotations.

3 Method

3.1 Preliminaries

We use conditional flow matching with the linear path $x_t = (1 - t)x_0 + tx_1$ between noise $x_0 \sim \mathcal{N}(0, I)$ and a clean latent video chunk x_1 . The model regresses the path velocity:

$$\mathcal{L}_{\text{flow}} = \mathbb{E} \left[\|v_\theta(x_t, t, c) - (x_1 - x_0)\|_2^2 \right]. \quad (1)$$

3.2 Action-Conditioned Flow

A pretrained video DiT defines a context-conditioned velocity field $v_\theta(x_t, t, c)$ whose integration produces a plausible continuation of the preceding frames c . For brevity, let $z_t = (x_t, t, c)$ denote the current latent state, flow time, and context. We expose a low-dimensional egomotion interface to redirect the flow by augmenting this field with an action a :

$$dx_t = v_\theta(z_t, a) dt. \quad (2)$$

The context z_t primarily determines the scene contents and plausible scene evolution while the action a directs scene motion, affecting both generation and scene evolution. Because egomotion influences generation in this way, we condition

the velocity network itself rather than applying a fixed transformation to the completed sample.

At a common latent state, we isolate and ground the action-induced change relative to a no-op command. Here and below, $\mathbf{0}$ denotes the semantic no-egomotion command:

$$\Delta v_\theta(z_t, a) = v_\theta(z_t, a) - v_\theta(z_t, \mathbf{0}). \quad (3)$$

The action representation determines the geometry that the conditional model is asked to learn. Rover egomotion is continuous and low dimensional: zero denotes no commanded motion, scalar changes represent command strength, different control axes may act simultaneously, and changing the sign of an axis requests the opposite motion. We therefore represent these actions in a centred affine coordinate system in which

$$\{0, \quad \lambda a, \quad a_1 + a_2, \quad -a\}$$

denote respectively no egomotion, scaled motion, compound motion, and motion in the opposite direction. We use this structure as an inductive bias. A generic categorical or learned latent action representation can in principle learn the same relationships, but the action space does not provide an intrinsic notion of neighbourhood, magnitude, addition, or inverse. Under limited data, those relationships must then be inferred separately.

The visual realization of a command nevertheless remains scene dependent. Depth, perspective, pose, visibility, occlusion, and scene geometry determine how the same egomotion changes the video. We therefore do not assume a fixed action-to-warp mapping. Instead, we hypothesize that, at a fixed z_t , the action-induced change in the velocity field is locally control-affine:

$$\Delta v_\theta(z_t, a) = G_\theta(z_t)a + \epsilon_\theta(z_t, a), \quad (4)$$

where G_θ is an implicit state-, time-, and scene-dependent action operator, and ϵ_θ captures action-dependent departures from this affine approximation. The nonlinear realization of motion is contained in G_θ ; the approximation concerns only how the low-dimensional command selects and combines these scene-dependent responses.

When ϵ_θ is small over the operational action range, the model approximately transfers the geometry of the command space to its velocity field:

$$\begin{aligned} \Delta v_\theta(z_t, \lambda a_1 + \gamma a_2) &\approx \lambda \Delta v_\theta(z_t, a_1) + \gamma \Delta v_\theta(z_t, a_2), \\ \Delta v_\theta(z_t, -a) &\approx -\Delta v_\theta(z_t, a), \\ v_\theta(z_t, 0) &\approx v_\theta(z_t). \end{aligned} \quad (5)$$

The first relation captures smooth scaling and composition: command magnitudes and combinations reuse common scene-dependent action directions rather than defining separate conditional behaviours. The second captures sign reversal: opposite egomotions occupy opposite sides of a shared axis. This in theory permits abundant motion on one side of an axis (forward motion, perhaps) to inform its shared response direction, while the scarcer opposite examples (backwards motion here) calibrate and test the negative branch. The zero command provides the common reference from which

every action-induced change is measured. Actions are therefore represented as redirections of one context-conditioned flow rather than as independent flow maps. Section 3.3 constructs the centred action coordinates, while our response, composition, and counterfactual evaluations test whether the learned generator transfers their geometry to the flow.

3.3 Data Derived Action Space

Section 3.2 motivates a centred command space with consistent notions of scale, composition, and sign. We recover such coordinates directly from unlabeled video motion, while leaving the scene-dependent visual realization of each command to the model.

In order to get fundamental movement signals, we use a pretrained off-the-shelf pixel tracking model [36] on a fixed 10×10 grid of starting pixels indexed by $j = 1, \dots, 100$. For every 0.5s chunk, represented by 3 latent frames spanning 12 pixel frames, the tracker follows and returns the net displacement $\delta = (\Delta x, \Delta y)$ for each grid point across the chunk.

$$m = [\delta_1^\top \quad \delta_2^\top \quad \dots \quad \delta_{100}^\top]^\top \in \mathbb{R}^{200}. \quad (6)$$

The chunk is long enough for coherent displacement to exceed tracker noise, but short enough that one approximately constant egomotion describes it and successive commands may be composed autoregressively. Examples are shown in Figure 1.

Let $\xi_{s,n} \in \mathbb{R}^2$ denote the unobserved planar egomotion of chunk n in scene s . Over a short interval, its stacked image-plane displacement can be locally represented as

$$m_{s,n} = J(c_{s,n}) \xi_{s,n} + \rho_{s,n} + \varepsilon_{s,n}, \quad (7)$$

where $J(c_{s,n})$ maps the rover egomotion into scene-dependent image motion, $\rho_{s,n}$ contains motion independent of the rover, and $\varepsilon_{s,n}$ contains tracking error and higher-order effects. The operator $J(c)$ accounts for depth, perspective, camera pose, foreshortening, visibility, and occlusion, and may vary substantially between scenes. Linearity is assumed only with respect to the short-horizon egomotion ξ at a fixed scene; image formation itself is not assumed to be globally linear.

Rover egomotion produces coherent global motion patterns that recur through the dataset. Independent pedestrians, vegetation, and tracker errors are more localized and vary across scenes. Pooling observations therefore emphasizes recurring egomotion structure while dispersing scene-specific residual motion.

Let $\bar{m} = \mathbb{E}[m]$ denote the training-set mean. We compute

$$C = \mathbb{E}[(m - \bar{m})(m - \bar{m})^\top], \quad (8)$$

$$Cu_k = \lambda_k u_k.$$

Here, λ_k is the variance captured by the k th principal component u_k . Although the scene-dependent operators $J(c)$ need not share an identical column space, the rover has only two commanded planar degrees of freedom and their global flow patterns recur across scenes. We therefore expect their contribution to concentrate in a small leading eigenspace when it dominates independent object motion and tracker

noise. This is a data-dependent assumption rather than a property guaranteed by PCA; we verify it through the explained variance and coherent flow structure of the recovered components.

Let $U = [u_1, \dots, u_d]$ contain the retained PCA directions. We project each tracked-motion vector into this basis as

$$p = U^\top(m - \bar{m}). \quad (9)$$

The resulting scores are scaled using their training-set variation and passed through a tanh function to obtain bounded actions $a \in (-1, 1)^d$. This preserves the centred, signed ordering of each PCA direction while reducing the influence of extreme motions. The resulting action space is locally linear near its centre and smoothly saturates toward its boundaries.

The PCA origin represents the mean tracked-motion field $\bar{m}$, not physical stationarity. Encoding an all-zero displacement field through the same pipeline gives

$$p_{\text{null}} = -U^\top \bar{m}, \quad a_{\text{null}} = (-0.0234, -0.0013). \quad (10)$$

Thus, the physical no-egomotion command is slightly offset from the numerical PCA origin. We use this correction only when explicitly requesting zero camera motion; all active training and evaluation commands retain their original encoded values. Throughout the geometric discussion, $\mathbf{0}$ denotes this semantic no-egomotion command.

The raw scores form affine coordinates in the retained subspace; after squashing, sign and ordering are preserved while scaling and composition are local because of saturation. On FrodoBots, the leading component captures forward-backward motion and explains approximately 60% of tracked-motion variance, while the second captures yaw and explains approximately 20%. Visualisations of the recovered basis are provided in Appendix B.

3.4 Latent Egomotion Critic without Decoder Gradients

Flow matching alone does not ensure that generated video realises the requested motion, so we supervise rendered egomotion with an online critic; Section 4.1 shows that removing the critic severely impairs command following. Let $y \in \mathbb{R}^8$ denote the retained PCA motion code and $a = y_{1:2} \in \mathbb{R}^2$ its throttle and yaw components. Only a is supplied to the DiT; the critic receives y and predicts all eight components to provide richer training supervision.

Direct supervision would require decoding, tracking, projecting, and differentiating through the full pixel pipeline, greatly increasing memory and potentially rewarding local pixel displacement rather than coherent scene-level egomotion. We therefore use the decoder and tracker as a frozen, no-gradient visual teacher.

At each training step, the current velocity prediction is converted into a one step clean latent estimate $\hat{x}_{\text{clean}}$ in pixel space. We apply the visual teacher to this estimate at every sampled flow timestep, without completing the generation trajectory or any time based filtering.

Let $\mathcal{T}_8$ denote the frozen pixel-space readout consisting of latent decoding, point tracking, displacement assembly,

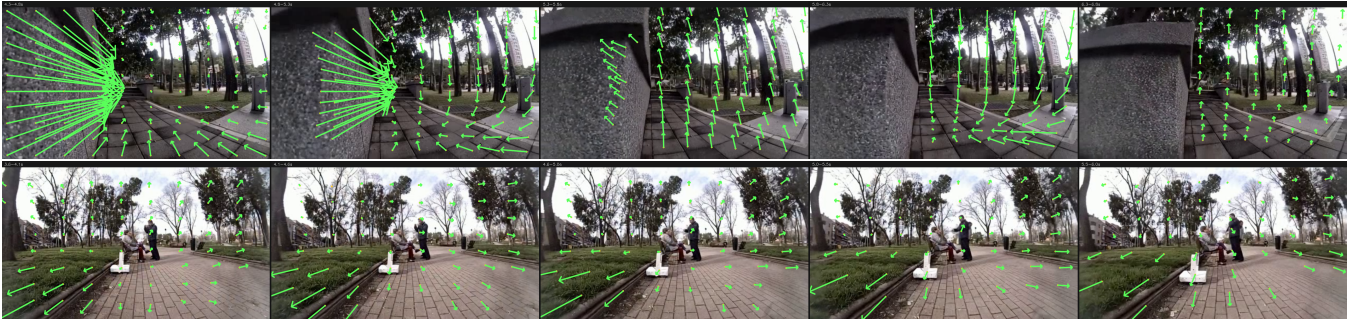


Figure 1: Tracked displacement fields from two real-world FrodoBots-2K sequences. Columns show successive 0.5 s chunks, and arrows show net grid-point displacement. Coherent egomotion patterns persist despite local motion and occlusion. Source video frames are from the FrodoBots-2K dataset, licensed under CC BY-SA 4.0; tracking overlays are ours.

PCA projection, scaling, and squashing from Section 3.3. The motion rendered by the current sample is measured as

$$y^{\text{teach}} = \mathcal{T}_8(\text{sg}(\hat{x}_{\text{clean}})) \in \mathbb{R}^8, \quad (11)$$

where $\text{sg}(\cdot)$ denotes stop-gradient. The supervisory label is therefore obtained through a full pixel-space motion measurement, but this pathway is forward-only: neither the decoder nor the tracker transmits gradients to the generator.

A lightweight critic D_ϕ learns to predict the same motion code directly from the same generated latent (in latent space, τ denotes the sampled flow timestep):

$$\hat{y}_\phi = D_\phi(\text{sg}(\hat{x}_{\text{clean}}), \tau, y). \quad (12)$$

The critic also receives the sampled flow timestep because the quality and statistics of $\hat{x}_{\text{clean}}$ vary with respect to diffusion time.

$$\mathcal{L}_{\text{crit}} = \|W_c(\hat{y}_\phi - y^{\text{teach}})\|_2^2. \quad (13)$$

The first two dimensions receive full weight because they correspond to the commanded throttle and yaw axes. The remaining six receive reduced weight, providing a richer description of rendered motion without expanding the two-dimensional control interface. Because the critic is trained online on samples from the generator before it is asked to supervise those same latents, its approximation is fresh and continuously on manifold.

$$\mathcal{L}_{\text{act}} = \|W_g(D_{\bar{\phi}}(\hat{x}_{\text{clean}}, \tau, y) - y)\|_2^2. \quad (14)$$

Here, $D_{\bar{\phi}}$ denotes the current critic with its parameters held fixed during the generator update and y is the PCA code of the real training chunk from which $a = y_{1:2}$ was obtained. The first two dimensions enforce the requested throttle and yaw, while the remaining six are lower-weight auxiliary targets for secondary camera and scene motion. Our `pca4` and `pca2` ablations show that reducing this auxiliary supervision degrades geometry and fine detail.

The complete generator objective is

$$\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{flow}} + \lambda_{\text{act}}\mathcal{L}_{\text{act}}, \quad (15)$$

Thus, the frozen teacher measures rendered motion in pixel space and the online critic transfers this signal to the latent

generator without decoder or tracker gradients. Both are used only during training and add no inference-time cost.

We train with teacher-forced chunks and generate autoregressively, allowing actions to change between chunks. Implementation details appear in Appendix C.

4 Evaluation

Setup We train on ~ 88 hours of FrodoBots video comprising 2,577 rides and $\sim 64\text{K}$ windows across eleven cities. Genuine reverse motion is exceptionally rare: the full $\sim 2,000$ -hour corpus contains only 632 sustained reverse windows (~ 55 minutes), which we oversample $6\times$. Training dynamics are analysed on four fixed training rides using both recorded commands (GT) and 180 degree rotated counterfactual commands (FLIP).

Each training example contains seven temporal latent chunks and is decomposed into seven teacher-forced next-chunk predictions, providing supervision at each autoregressive transition. We train all variants for 5,000 optimiser steps. Global batch sizes of 16, 32, and 64 correspond to approximately 1.25, 2.5, and 5 dataset-equivalent epochs, respectively. Unless otherwise stated, `Default` uses a global batch size of 32 and eight PCA components for critic supervision; only the leading two components are exposed as throttle and yaw input controls.

For cross-model evaluation, we use 32 low-egomotion contexts from held-out rides, each continued under eight constant compass commands $\uparrow$ (forward), $\downarrow$ (back), $\leftarrow$ (left), $\rightarrow$ (right), $\nearrow$ (forward right), $\searrow$ (back right), $\swarrow$ (back left), $\nwarrow$ (forward left), giving $32 \times 8 = 256$ directional rollouts per model. Every model is rolled out over a common six-second horizon at its native frame rate and through its native autoregressive interface. We additionally generate one no-op continuation from each context, giving 32 stationary rollouts per model; these are evaluated separately from the directional fleet.

Comparisons. We compare against state-of-the-art WMs including minWM [20], WorldCam [21], Astra [22], Matrix-Game 2.0 [23], HY-World 1.5 WorldPlay [24], and Yume-1.5 [25]. Each compass command is translated into the model’s native pose, keyboard–mouse, or text interface and

held constant throughout the rollout. Because these interfaces differ in units and permitted command strengths, we compare realized direction, forward–backward asymmetry, and translation–steering composition rather than absolute control gain. Full benchmark construction and baseline control mappings are provided in Appendix D, with model-specific interfaces detailed in Section D.2. We ablate conditioning pathways, critic supervision, supervision width, and batch size. No AdaLN retains only action tokens; No Act Tok retains only AdaLN action conditioning. No Critic retains both pathways but trains the generator with $\mathcal{L}_{\text{flow}}$ alone. pca4/pca2 reduce critic supervision from eight to four/two PCA components, leaving the two-dimensional control input unchanged. Batch16/Batch64 use global batch sizes of 16/64 instead of the default 32.

4.1 Controllability

We measure every model with the same frozen CoTracker-PCA readout, which projects fixed-grid displacements into signed throttle-yaw coordinates for forward, backward, steering, and compound commands, allowing them to be evaluated within one representation. Existing image-based camera estimators cannot provide this comparison reliably: they generally assume a rigid projective transform and become unreliable when generated scenes deform or when several motion axes must be measured simultaneously.

Directional control is evaluated with two complementary tests. CoTracker-PCA measures whether realised motion follows the commanded direction, while ORB feature matching with RANSAC homography verification detects near-stasis. Matching uses 3000 keypoints at 640×352 resolution and a 5-pixel homography tolerance; a rollout is treated as near-static when more than 500 verified correspondences remain between its first generated frame and its six-second frame.

The same readout also supplies our training labels, but the resulting advantage is strictly bounded: evaluation is performed using a fixed, non-parametric pipeline applied unchanged to every model. Training on this space does not guarantee that a model will learn it. Moreover, the readout requires only unlabelled video and remains available equally to other world-model systems as either a supervision signal or an evaluation instrument.

GT continuation can masquerade as control. Figure 2 compares recorded commands (GT) with counterfactual commands obtained by reversing both action axes (FLIP). GT agreement is high early because the pretrained model continues motion from the context. Backward motion is the exception: although reverse errors are penalised by the squared loss, reverse examples are so rare that the aggregate objective can improve while this branch remains underfit (Fig. 10). FLIP instead tests whether the command redirects that continuation: steering emerges quickly, whereas forward–backward control appears later. The diagonal responses show that throttle and steering are composed rather than learned as unrelated categories. Extended training curves, conditioning ablations, settled responses, and additional cross-model analyses appear in Appendix E.

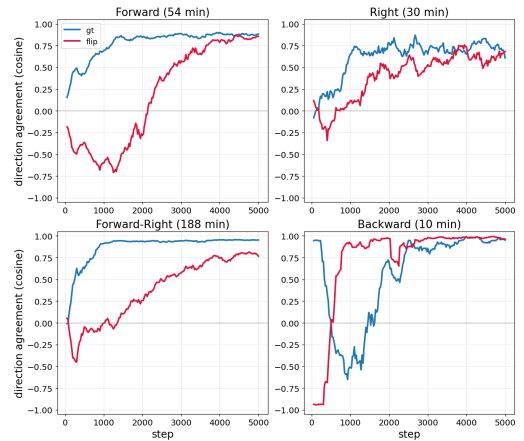


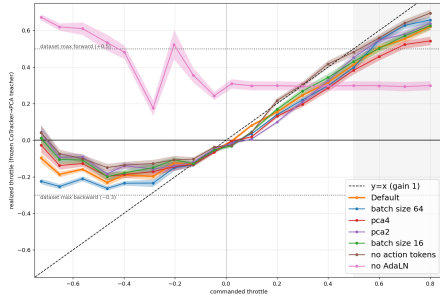
Figure 2: Training-time cosine agreement for recorded (GT) and sign-reversed (FLIP) commands. Positive FLIP agreement indicates genuine command response; panel titles give source-direction duration in minutes. The full eight-direction result appears in Fig. 10.

Response follows the data support. Figure 3 shows approximately linear steering, while throttle saturates outside the observed training range. As batch size or the number of critic-supervised PCA dimensions is reduced, the reverse curves also bend back toward zero at progressively larger negative command magnitudes. This highlights a limitation of our data-derived approach: response strength depends strongly on the coverage of the observed motion distribution. Conversely, the absence of the same contraction on the forward branch is consistent with its much denser representation in the training data. Nevertheless, every functional variant recovers the sign of backward motion despite its extreme scarcity.

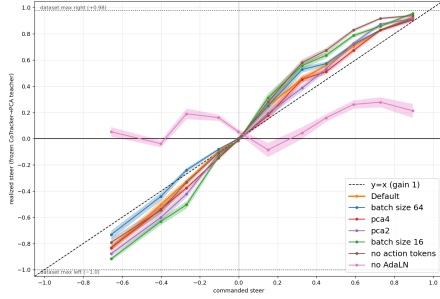
Backward composition distinguishes the models. Figure 4 applies the same eight-direction action-forcing test to all models. Some baselines (Astra, WorldCam) fail to produce backward motion at all. Others reverse under the pure backward command but lose or flip that component when steering is added, showing that backward, backward-left, and backward-right are not represented compositionally. Our model also reflects the severe forward–backward data imbalance, producing asymmetric wedges, but it gives the strongest and most consistent reverse response across both pure and diagonal commands. minWM also preserves the correct structure, but with substantially weaker realized motion.

Control failure by command. Table 1 shows that backward control is the main failure mode for WorldCam, Astra and Yume, which fail 93%, 87% and 40% of backward commands respectively. In contrast, minWM struggles across all directions, with 84/87 control failures caused by near-stasis.

Table 2 meanwhile shows that removing AdaLN or critic supervision raises overall control failure from 3% to 58% and 88%, respectively, demonstrating that both are critical for reliable action following. The remaining ablations retain low



(a) Forward and backward response.



(b) Steering response.

Figure 3: Settled throttle and steering responses of the ablation variants, measured by the CoTracker-PCA readout. Most points use sign-flipped commands on held-out windows; positive throttle instead uses sampled forward commands on held-out no-op contexts. Dashed lines mark the approximate training support.

overall failure (4–8%), but reduced PCA supervision, smaller batches and removal of action tokens weaken reversal: backward failure reaches 27–37%, versus 7% for Default and 10% for Batch64.

4.2 Quality

Action-forced rollouts do not have paired ground-truth continuations, making standard reference-based video metrics unsuitable. We therefore evaluate five complementary failure signals: *style shift* (changes in rendering appearance), *geometric corruption* (distorted or implausible scene structure), *scene relocation* (loss of the original place identity), *conjunction* (persistent objects appearing without a plausible visual history), and *progressive high-frequency degradation* (loss of fine detail over time). Each instrument is selected for its agreement with human judgement. Style and geometry are assessed against real context frames, while conjunction is detected from within-rollout object histories. Only HF and relocation use peers: HF normalises sharpness loss by the same-input peer median to account for scene difficulty, while relocation uses peer endpoints alongside real context frames as alternative views for checking place identity. The reference contains one representative per family, with each ablation replacing Default when scored. Metric construction is described in Appendix F, and validation against human annotations is reported in Appendix G.

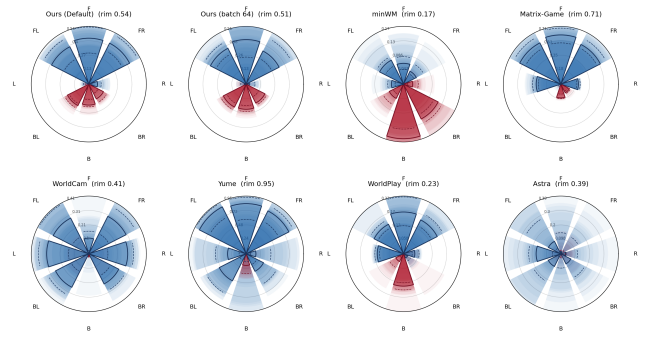


Figure 4: Realised throttle g_0 under eight commands. Each wedge is a collation of one direction; opacity shows density, solid outlines medians, and dashed arcs interquartile ranges. Blue denotes forward and red backward motion. Panels use separate 95th-percentile rims, so compare sign and shape rather than absolute radius.

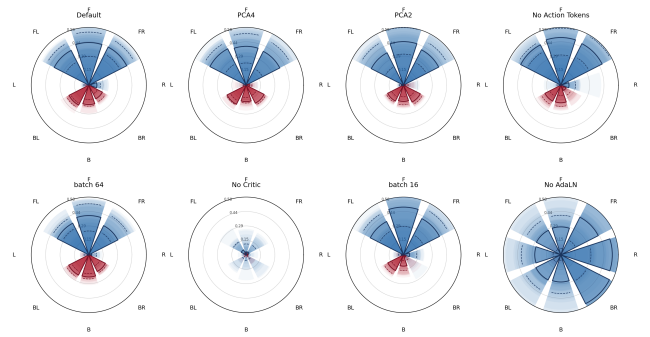


Figure 5: Wedge plots for the ablation suite. The No Critic and the No AdaLN model exhibit markedly weaker and less coherent realised responses than the other models, showing that control performance is strongly dependent on these axes.

Table 3 shows distinct failure profiles. Yume preserves style and geometry but frequently relocates, while minWM is visually stable but often freezes or conjures objects. Our Default model records 3% style shift, 17% geometric corruption, 20% relocation, 1% conjunction, and 16% high-frequency degradation. Reducing critic supervision to pca4/pca2 raises geometric corruption from 17% to 38%/40%, while removing action tokens increases HF degradation from 16% to 29%. Smaller batches also lose detail: HF degradation is 44% for Batch16 versus 14% for Batch64. Compared with Default, Batch64 reduces relocation (20% to 17%) and HF degradation (16% to 14%), but increases geometric corruption (17% to 26%). Conversely, No AdaLN/No Critic reduce geometric corruption to 11%/2%, and No Critic lowers relocation to 5%, but these gains in individual quality metrics accompany poor action following (Table 4).

Full degradation analyses are reported in Appendix F.6.

Command	Default	WorldPlay	Matrix-Game	WorldCam	Astra	Yume	minWM
↑	7	13	7	23	67	3	67
↓	7	3	0	93	87	40	60
←	3	0	0	0	7	0	23
→	3	0	0	0	3	3	30
↖	7	0	0	3	3	0	27
↗	0	0	3	0	3	0	27
↘	0	0	0	60	43	63	27
↙	0	0	0	77	33	63	30
All	3	2	1	32	31	22	36

Table 1: Control failure by command (%) for our flagship and the external baselines; lower is better.

Command	Default	pca4	pca2	Batch64	Batch16	No Act Tok	No AdaLN	No Critic
↑	7	3	10	17	7	7	47	87
↓	7	27	33	10	27	37	97	100
←	3	0	0	0	0	3	77	97
→	3	0	0	3	0	0	13	80
↖	7	3	7	7	7	7	40	77
↗	0	0	0	0	0	0	7	77
↘	0	0	10	7	0	3	90	97
↙	0	0	7	0	3	0	93	87
All	3	4	8	5	5	7	58	88

Table 2: Control failure by command (%) for the ablation grid, with the same definitions and populations as Table 1.

4.3 Stationarity

A correct no-op should hold the viewpoint without freezing independent scene dynamics. We evaluate the same 32 contexts under stationary commands, flagging rollouts whose CoTracker-PCA action magnitude reaches 0.1 in squashed units. We use the motion readout rather than frame matching because geometric corruption can break correspondences without moving the viewpoint. Stationary quality and residual animation are evaluated separately in Appendix H.

Thus, directional control does not automatically extend to stationarity. Despite receiving stationary commands, Astra, Yume and WorldCam move in 75%, 44% and 22% of rollouts, respectively (Table 5). Default and all functional ablations have no flagged no-op rollouts, whereas removing AdaLN raises failure to 78% (Table 6). Taken together with minWM’s static failures under motion commands, these results highlight the value of a grounded, affine action space: it supports interpolation between observed actions, enabling both a calibrated no-op and composable directional control within the training support.

Model	Style %	Geom. %	Reloc. %	Conj. %	HF %
WorldPlay	16	80	78	0	13
Matrix-Game	62	71	96	0	9
WorldCam	62	68	88	0.4	45
Astra	13	52	68	0.4	11
Yume	5	9	59	0.4	13
minWM	4	1	4	20	5
Ours (Default)	3	17	20	1	16

Table 3: Generation-quality failure rates (%); lower is better.

Model	Style %	Geom. %	Reloc. %	Conj. %	HF %
Default	3	17	20	1	16
pca4	3	38	28	0	22
pca2	3	40	25	0	28
Batch64	3	26	17	0	14
Batch16	5	32	42	1	44
No Act Tok	3	29	31	1	29
No AdaLN	4	11	30	2	53
No Critic	0	2	5	0	23

Table 4: Generation-quality failure rates (%) for the ablations. Each variant replaces Default as the sole family representative when scored.

4.4 Combined Metrics

We call a rollout *legitimate* when it passes the control criterion and all five quality tests.

Figures 6a and 6b reveal distinct failure profiles. minWM preserves visual quality comparatively well, but frequent control failures (black) and conjuration (green) limit its usefulness as a controllable world model. Geometric corruption (red) dominates the displayed failures of Matrix-Game, WorldPlay and WorldCam, while relocation (blue) dominates Yume; Astra combines geometry and control failures. Default achieves the highest joint pass rate, represented by light-grey cells, although HF degradation (orange) remains a notable source of error.

These failure profiles also motivate our use of real-world video. We hypothesise that third-person gameplay can encourage foreground-actor priors underlying conjuration, while training on generated video can propagate rendering biases and style shifts across successive models. Learning from noisy, low-resolution footage remains challenging, and our model still exhibits HF degradation. However, these deployment challenges must ultimately be addressed; synthetic-only training risks introducing additional biases rather than resolving them.

5 Conclusion & Limitations

Learning grounded actions from unlabelled video is challenging because observed motion entangles camera egomotion with scene geometry, independent object motion, and noise, yet solving this problem is essential for scaling world models beyond costly instrumented datasets. We show that using a simple pixel tracker to derive pixel vector fields and finding their principal components results in recovering a re-

	Default	WorldPlay	Matrix-Game	WorldCam	Astra	Yume	minWM	REAL
No-op fail (%)	0	0	0	22	75	44	3	6

Table 5: No-op control failure (%) over 32 contexts, using the 0.1 action-magnitude threshold defined in the text. REAL reports false positives from noisy action labelling on real no-op continuations, not a lower bound on model failure rates.

	Default	pca4	pca2	Batch64	Batch16	No Act Tok	No AdaLN	No Critic
No-op fail (%)	0	0	0	0	0	0	78	3

Table 6: Stationary control failure (%) for the ablation grid, with the same definition as Table 5.

versible, composable, and linear action space with which we learn reliable backward control and composable steering despite scarce backward footage. Crucially, this approach can exploit the scale and diversity of real-world footage rather than restricting world-model training to simulated environments or videos collected within games.

Our method is both strengthened and limited by its core data dependency - any axes of egomotion present in the data can be recovered but motion not present in the data cannot be learnt in this manner. We also find that while our model interpolates well within the support provided (and better than the other baselines), extrapolation outside of the given support is somewhat lacking as shown by Fig 3a.

Action forcing evaluation also shows that conventional evaluation of reference-based quality is insufficient as a metric to evaluate world models. Our controllability wedges and quality metrics offer one potential manner in which WM evaluation can be done.

As a final takeaway, we wish to highlight that world model training no longer needs to be bounded by the quality of associated action labels.

Acknowledgements

This work made use of the Isambard high-performance computing facility. We thank the FrodoBots team for making the FrodoBots-2K dataset available under the CC BY-SA 4.0 licence.

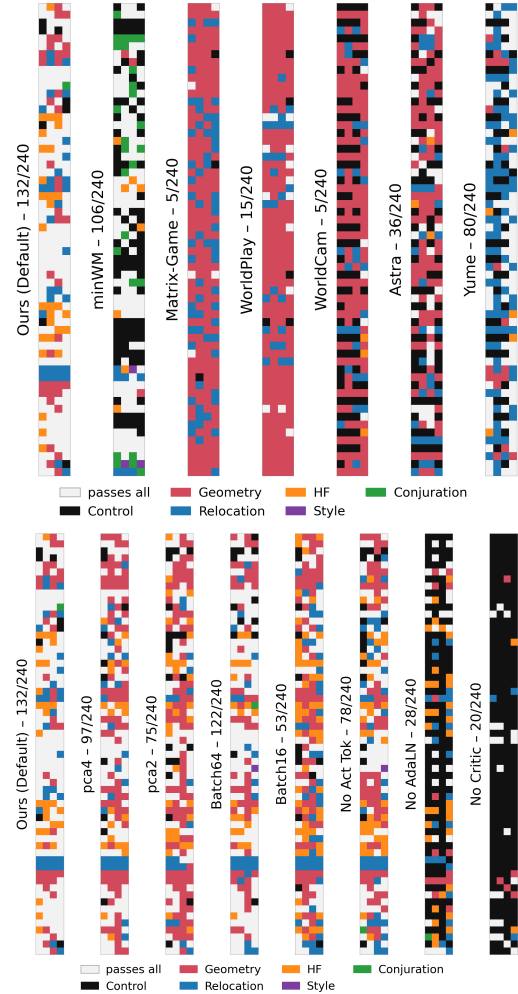


Figure 6: Per-rollout failure profiles for (a) Default and external baselines and (b) our ablations. Neutral cells pass all criteria; coloturs show the first failure in priority order: control > geometry > relocation > HF > style > conjunction. Headings report the number passing all criteria.

References

- [1] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15791–15801, June 2025.
- [2] Hao He, Ceyuan Yang, Shanchuan Lin, Yinghao Xu, Meng Wei, Liangke Gui, Qi Zhao, Gordon Wetzstein, Lu Jiang, and Hongsheng Li. Cameractrl ii: Dynamic scene exploration via camera-controlled video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13416–13426, October 2025.
- [3] Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.
- [4] Yifei Dong, Fengyi Wu, Guangyu Chen, Lingdong Kong, Xu Zhu, Qiyu Hu, Yuxuan Zhou, Jingdong Sun, Jun-Yan He, Qi Dai, Alexander G. Hauptmann, and Zhi-Qi Cheng. Towards unified world models for visual navigation via memory-augmented planning and foresight. *arXiv preprint arXiv:2510.08713*, 2025.
- [5] Chaoda Zheng, Sean Li, Jinhao Deng, Zhennan Wang, Shijia Chen, Liqiang Xiao, Ziheng Chi, Hongbin Lin, Kangjie Chen, Boyang Wang, et al. X-world: Controllable ego-centric multi-camera world models for scalable end-to-end driving. *arXiv preprint arXiv:2603.19979*, 2026.
- [6] Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.
- [7] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations (ICLR)*, 2025.
- [8] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.
- [9] Yuxin Jiang, Yuchao Gu, Ivor W Tsang, and Mike Zheng Shou. Olaf-world: Orienting latent actions for video world modeling. *arXiv preprint arXiv:2602.10104*, 2026.
- [10] Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. Mineworld: a real-time and open-source interactive world model on minecraft. *arXiv preprint arXiv:2504.08388*, 2025.
- [11] Jiwen Yu, Yiran Qin, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Gamefactory: Creating new games with generative interactive videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11590–11599, 2025.
- [12] Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory. *Advances in Neural Information Processing Systems*, 38:49632–49652, 2026.
- [13] Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C. Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 4603–4623. PMLR, 2024.
- [14] Shenyan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. *arXiv preprint arXiv:2503.18938*, 2025.
- [15] Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, et al. Relic: Interactive video world model with long-horizon memory. *arXiv preprint arXiv:2512.04040*, 2025.
- [16] FrodoBots Lab, Michael Cho, Sam Cho, Aaron Tung, Nireesh Dravin, and Santiago Pravisani. FrodoBots-2K Dataset. <https://huggingface.co/datasets/frodobots/FrodoBots-2K>, 2024. Licensed under CC-BY-SA-4.0.
- [17] Wonjoon Jin, Qi Dai, Chong Luo, Seung-Hwan Baek, and Sunghyun Cho. Flovd: Optical flow meets video diffusion model for enhanced camera-controlled video synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2040–2049, 2025.
- [18] Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Yusuf Aytar, Michael Rubinstein, Chen Sun, et al. Motion prompting: Controlling video generation with motion trajectories. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1–12, 2025.
- [19] Mathis Koroglu, Hugo Caselles-Dupré, Guillaume Jeanneret, and Matthieu Cord. Onlyflow: Optical flow based motion conditioning for video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6226–6236, 2025.
- [20] Min Zhao, Hongzhou Zhu, Bokai Yan, Zihan Zhou, Yimin Chen, Wenqiang Sun, Kaiwen Zheng, Guande He, Xiao Yang, Chongxuan Li, et al. minwm: A full-stack open-source framework for real-time interactive video world models. *arXiv preprint arXiv:2605.30263*, 2026.
- [21] Jisu Nam, Yicong Hong, Chun-Hao Paul Huang, Feng Liu, JoungBin Lee, Jiyoung Kim, Siyoon Jin, Yunsung Lee, Jaeyoon Jung, Suhwan Choi, et al. Worldcam:

- Interactive autoregressive 3d gaming worlds with camera pose as a unifying geometric representation. *arXiv preprint arXiv:2603.16871*, 2026.
- [22] Yixuan Zhu, Jiaqi Feng, Wenzhao Zheng, Yuan Gao, Xin Tao, Pengfei Wan, Jie Zhou, and Jiwen Lu. Astra: General interactive world model with autoregressive denoising. *arXiv preprint arXiv:2512.08931*, 2025.
 - [23] Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, et al. Matrix-game 2.0: An open-source real-time and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025.
 - [24] Team HunyuanWorld. Hy-world 1.5: A systematic framework for interactive world modeling with real-time latency and geometric consistency. *arXiv preprint*, 2025.
 - [25] Xiaofeng Mao, Shaoheng Lin, Zhen Li, Chuanhao Li, Wenshuo Peng, Tong He, Jiangmiao Pang, Mingmin Chi, Yu Qiao, and Kaipeng Zhang. Yume: An interactive world generation model. *arXiv preprint arXiv:2507.17744*, 2025.
 - [26] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Bajaj, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.
 - [27] Haoran He, Yang Zhang, Liang Lin, Zhongwen Xu, and Ling Pan. Pre-trained video generative models as world simulators. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 4645–4653, 2026.
 - [28] Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2world: Crafting video diffusion models to interactive world models. *arXiv preprint arXiv:2505.14357*, 2025.
 - [29] Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *International Conference on Learning Representations*, volume 2024, pages 9379–9395, 2024.
 - [30] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, volume 2025, pages 28213–28239, 2025.
 - [31] Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhui Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12268–12290, 2024.
 - [32] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pages 366–384. Springer, 2024.
 - [33] Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. Videophy: Evaluating physical commonsense for video generation. In *International Conference on Learning Representations*, volume 2025, pages 102075–102121, 2025.
 - [34] Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhramil Chandra, Ziyang Jiang, Aaran Arulraj, et al. Videoscore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2105–2123, 2024.
 - [35] Hui Han, Siyuan Li, Jiaqi Chen, Yiwen Yuan, Yuling Wu, Yufan Deng, Chak Tou Leong, Hanwen Du, Junchen Fu, Youhua Li, et al. Video-bench: Human-aligned video generation benchmark. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18858–18868, 2025.
 - [36] Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6013–6022, October 2025.
 - [37] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.
 - [38] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.
 - [39] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, 2024.
 - [40] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Dets beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16965–16974, 2024.
 - [41] Lingzhi Zhang, Zhengjie Xu, Connelly Barnes, Yuqian Zhou, Qing Liu, He Zhang, Sohrab Amirghodsi, Zhe Lin, Eli Shechtman, and Jianbo Shi. Perceptual artifacts localization for image synthesis tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7579–7590, 2023.
 - [42] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Wei-jie Su, Jie Shao, et al. Internv13: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

- [43] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.
- [44] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- [45] Priya Goyal. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

A Example Generations

Representative continuations seeded with approximately 0.5 seconds of real video are shown in Figure 7.

B Recovered Motion Basis

Figure 8 visualizes the leading motion directions recovered by the fixed CoTracker-PCA pipeline. The first component, g_0 , captures longitudinal forward-backward motion; g_1 captures yaw; and g_6 captures roll-like camera motion.

C Implementation Details

Each training video contains seven chunks and we train using the teacher forcing paradigm [37]. We decompose the seven chunk sequence into seven teacher-forced prediction sequences of varying lengths, each targeting one chunk with a different amount of preceding ground-truth context.

When inference is run with 48 diffusion steps like in training, it takes 86s (on a GH200) to generate 7 chunks (or roughly 6 seconds) of content autoregressively, which at 16 FPS gives a generation speed of 1.11 FPS. Since Wan is a flow model, 20 diffusion steps suffice to generate with minimal loss in quality which subsequently increases the generation FPS to 2.72 FPS.

Critic conditioning and targets. The training configurations use PCA components 0 and 1 for throttle and steering and components 0–7 for the critic; in Section 3.4, these are indexed as $a = y_{1:2}$ and $y \in \mathbb{R}^8$. The target y is the real training window’s per-frame PCA projection, scaled and tanh-squashed componentwise, then mean-pooled within the target chunk. Only a conditions the DiT. The critic receives the full y , the latent estimate and the sampled flow timestep in both updates.

During the critic update, its target is y^{teach} , obtained by applying the frozen decoder-tracker-PCA teacher to the detached generated latent. It is therefore trained to predict rendered motion, not to copy its conditioning vector. During the generator update, the critic parameters are frozen and its outputs are regressed onto the real-window target y :

$$\begin{aligned}\hat{y} &= D_{\hat{\phi}}(\hat{x}_{\text{clean}}, \tau, y), \\ \mathcal{L}_{\text{act}} &= \text{MSE}(\hat{y}_{1:2}, y_{1:2}) \\ &\quad + 0.3333 \text{MSE}(\hat{y}_{3:8}, y_{3:8}).\end{aligned}$$

The first term supervises the commanded throttle and steering; the second uses the real continuation’s six auxiliary motion components. Both use the same PCA basis and tanh scales as the teacher. The matrix W_g in Eq. 14 incorporates these weights and mean reductions. The overall coefficient is $\lambda_{\text{act}} = 0.3$ in Eq. 15 and 0 for No Critic; it is applied once.

D Cross-Model Evaluation Protocol

D.1 Action-Forcing Benchmark

Low-egomotion context selection. The benchmark uses 32 contexts drawn from distinct held-out FrodoBots rides.

For each candidate window w , we compute the mean ego-motion magnitude across its N_w chunks:

$$e(w) = \frac{1}{N_w} \sum_{t=1}^{N_w} \sqrt{a_{t,\text{throttle}}^2 + a_{t,\text{yaw}}^2},$$

where $a_{t,\text{throttle}}$ and $a_{t,\text{yaw}}$ are the tanh-squashed PCA action coordinates for chunk t .

Let q_{25} denote the 25th percentile of the ego-motion scores across all candidate windows. We define the eligibility threshold as

$$\tau_{\text{stat}} = \min(0.12, q_{25}),$$

and retain candidate windows satisfying

$$e(w) \leq \tau_{\text{stat}}.$$

This selects at most the lowest-motion quarter of the candidate population while also imposing an absolute upper limit of 0.12 in squashed PCA units.

The selection criterion restricts camera ego-motion rather than all visible motion. A context may still contain independently moving pedestrians, vehicles, vegetation, or other scene elements. A separate point-track scene-activity score is used elsewhere for evaluation but is not part of this low-egomotion selection criterion.

For our model, actions are ordered as

$$a = (a_{\text{throttle}}, a_{\text{yaw}}).$$

All commands have Euclidean norm $s = 0.5$. The cardinal commands are

$$\uparrow = (s, 0), \quad \downarrow = (-s, 0), \quad \rightarrow = (0, s), \quad \leftarrow = (0, -s),$$

and the diagonal commands are

$$\begin{aligned}\nearrow &= \frac{s}{\sqrt{2}}(1, 1), & \nwarrow &= \frac{s}{\sqrt{2}}(1, -1), \\ \swarrow &= \frac{s}{\sqrt{2}}(-1, 1), & \searrow &= \frac{s}{\sqrt{2}}(-1, -1).\end{aligned}$$

Thus, each active diagonal coordinate has magnitude $s/\sqrt{2} \approx 0.354$. Normalizing the diagonals in this way tests translation-steering composition without increasing total command magnitude relative to the cardinal directions.

Context and rollout duration. Our model receives one temporal context chunk containing three latent frames, corresponding to 12 RGB frames or 0.75 s at 16 fps. It then generates eight autoregressive chunks, corresponding to 96 RGB frames or 6.0 s. The complete context-plus-generation clip therefore contains 108 RGB frames and spans 6.75 s.

Sampling and reproducibility. Our model uses the rectified flow-matching sampler with 48 denoising steps per generated chunk, `timestep_shift= 5.0`, and `real_guidance_scale= 0.0`, corresponding to no classifier-free guidance. We evaluate the step-5,000 checkpoint `causal_lora_step0005000.pt`. The same sampling configuration is used for all of our ablations.

Sampling is deterministic. For generated block $b \in \{0, \dots, 7\}$, the initial noise is drawn using

$$\text{seed}(b) = 1234 + b.$$



Figure 7: Representative rollouts across environments and cities. The commanded action is shown in the leftmost panel of each row.

The same fixed eight-block noise sequence is reused for every context and command, holding sampling noise constant across the action-forcing comparisons.

Each baseline uses its released evaluation checkpoint, sampler, inference-step count, guidance configuration, and random-seed handling without modification. The only extensions are the control-interface mappings described in Section D.2.

Evaluation resolution and motion readout. Each model is evaluated at its native output resolution; no shared resizing is applied before the CoTracker-PCA controllability readout. The resolutions are listed in Table 7.

Because the conditioning interfaces and native output resolutions are not metrically matched, raw PCA response magnitudes are not treated as directly comparable across models. Cross-model analysis instead emphasizes response sign, requested direction, within-model forward-backward asymmetry, and translation-steering composition. For the separate cross-model image-translation analysis, only width-normalized horizontal and vertical translations are compared.

Model	Output resolution
Ours	832×480
minWM	832×480
WorldCam	832×480
HY-World 1.5 WorldPlay	832×480
Astra	832×480
Matrix-Game 2.0	640×352
Yume-1.5	1280×704

Table 7: Native output resolutions used by the action-forcing benchmark.

Failed generations. Each context-command rollout is executed independently. If generation fails, the rollout is skipped entirely: no video or metrics row is written, and the missing result is not zero-filled or replaced. In practice, after the control-interface implementations were finalized, all 256 rollouts completed successfully for every evaluated model. No benchmark rows were therefore removed because of generation failure.

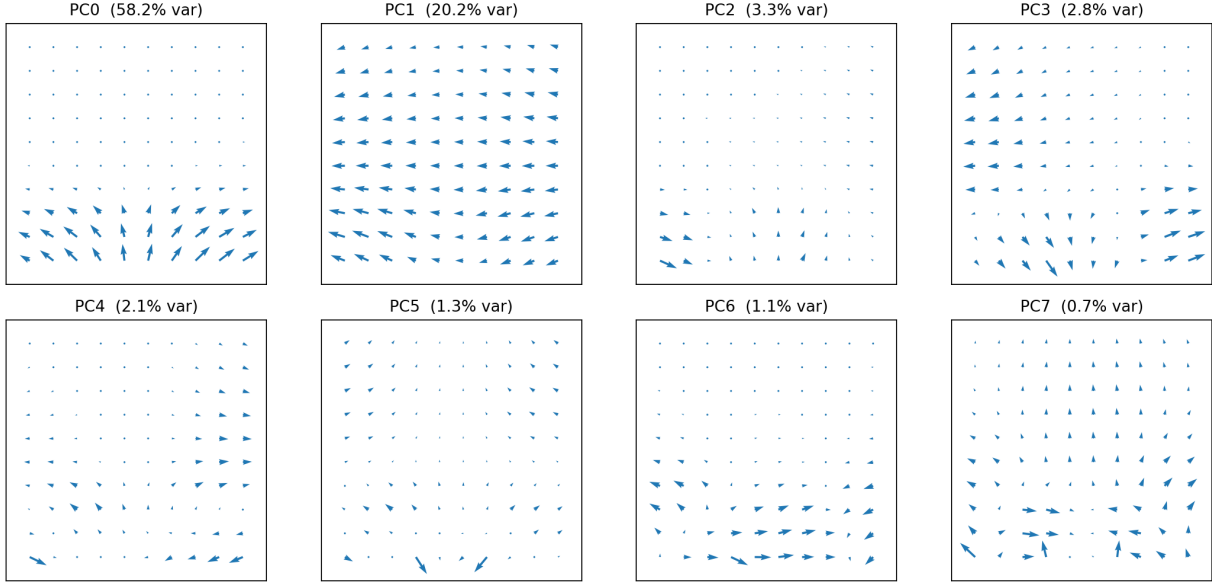


Figure 8: Principal motion components recovered from the fixed CoTracker-PCA pipeline. Each panel shows the displacement field associated with one PCA coordinate. The exposed control space uses g_0 for longitudinal motion and g_1 for yaw; g_6 captures roll-like motion. The remaining components describe secondary camera and scene dynamics and are used only for critic supervision.

D.2 Baseline Control Interfaces

Common protocol. Each compass command is translated into the closest operation supported by the released model and repeated throughout generation. For pose-conditioned models, the running camera pose is composed with a fixed relative translation and/or yaw increment. For keyboard-mouse and text-conditioned models, the corresponding control input or caption is held constant throughout the rollout. For a given model, all parameters are fixed across scenes and directions.

minWM minWM accepts an autoregressive sequence of relative SE(3) camera poses. We use the released trajectory-generator increments of 0.08 translation and 3° yaw per generation step. Forward and backward use positive and negative longitudinal translation, left and right use signed yaw, and diagonal commands combine the corresponding translation and yaw increments. The no-op command uses zero translation and zero yaw, leaving the relative pose at identity for every generation step.

HY-World 1.5 WorldPlay WorldPlay is also driven using relative SE(3) camera poses and uses the released trajectory scale of 0.08 translation and 3° yaw per step. Its pose-string interface cannot express simultaneous translation and yaw, so diagonal commands are supplied through the released JSON pose interface. Backward and backward-diagonal commands use negative longitudinal translation. The no-op command supplies an identity extrinsic for every frame through the same JSON pose interface.

Astra Astra conditions on a per-frame relative camera pose represented by a flattened 3×4 matrix. We follow the

scale used in the released demonstration code, including its translation increment of 0.03. The release provides forward and signed-steering primitives. We construct backward motion by reversing the longitudinal translation and construct backward-diagonal commands by combining this translation with the released signed-yaw primitive. The no-op command holds the identity relative pose $\mathbf{I}_{3 \times 4}$ for every frame. This matches the release’s own convention: its trajectory generator already assigns identity poses to the conditioning frames, annotated as zero-motion camera pose, and our no-op extends that convention across the generated span rather than switching to a translation increment at the hand-off.

WorldCam WorldCam is conditioned on per-pixel-frame camera-to-world trajectories in OpenCV convention. Its pose scale is inherited from its training trajectories and is not metric. Because the release does not include an eight-direction trajectory generator, we construct one through the released pose interface.

The conditioning span uses identity poses. During generation, the running pose is composed multiplicatively as

$$C_{t+1} = C_t D,$$

where D contains a longitudinal translation of

$$\Delta z = \pm 0.02$$

camera-to-world units per pixel frame and/or a yaw rotation of $\pm 1.5^\circ$ about the y -axis. Positive and negative translations are used for the forward and backward command families, respectively. Positive yaw is used for right-family commands and negative yaw for left-family commands. Diagonal commands combine the corresponding translation and yaw within the same increment.

Command	Keyboard input	Mouse yaw
↑	forward=1	0
↓	backward=1	0
→	none	+0.1
←	none	-0.1
↗	forward=1	+0.1
↖	forward=1	-0.1
↘	backward=1	+0.1
↙	backward=1	-0.1
∅	none	0

Table 8: Matrix-Game 2.0 controls used for the eight-direction benchmark.

The released runner performs 50 autoregressive steps and produces 200 generated frames. We use the fixed intrinsics

$$f_x = f_y = 723, \quad c_x = 960, \quad c_y = 540.$$

The translation and yaw increments were fixed during interface calibration before the action-forcing benchmark and were then applied unchanged across every scene and direction. The no-op command sets $D = I$, that is $\Delta z = 0$ with zero yaw, so the running pose $C_{t+1} = C_t D$ remains at the identity of the conditioning span throughout generation.

Matrix-Game 2.0 Matrix-Game accepts a six-dimensional binary keyboard stream and a two-dimensional mouse stream, both held constant throughout generation. We use its released benchmark mouse-yaw magnitude of 0.1 per pixel frame. The eight command mappings are given in Table 8.

Keyboard controls are binary, whereas mouse yaw has a continuous magnitude. Backward input is supported by the released interface but is not exercised in the model’s released benchmark protocol.

Yume-1.5. Yume is controlled through natural-language captions drawn from its supported training vocabulary. Each command uses the common prefix

This video depicts a city walk scene with a first-person view (FPV).

followed by the corresponding person and camera clauses in Table 9. The complete caption is repeated throughout the rollout. Yume exposes directional commands but no numerical command magnitude and therefore cannot be evaluated using a continuous dose sweep.

Text-conditioning neutrality. Several baselines accept a text prompt alongside their action interface. Under a directional command, motion-referring language in that prompt is harmless; under a no-op it contradicts the commanded behaviour. Three of the prompts we used for the directional benchmark contain such language: Astra’s scene prompt ends with “*The camera moves smoothly through the scene*”; Yume’s common prefix describes a “*city walk*” scene; and WorldCam inherits its base model’s default negative prompt, which lists three terms meaning *static*, *motionless*, and *a still, unmoving picture* among the artefacts to be avoided, thereby penalising the behaviour under test.

Command	Person clause	Camera clause
↑	moves forward (W)	remains still
↓	moves backward (S)	remains still
→	stands still	turns left
←	stands still	turns right
↖	moves forward and left (W+A)	turns left
↗	moves forward and right (W+D)	turns right
↙	moves backward and left (S+A)	turns left
↘	moves backward and right (S+D)	turns right
∅	stands still (·)	remains still (·)

Table 9: Yume-1.5 caption clauses used for the eight-direction benchmark.

For the stationary evaluation we therefore repeat these rollouts with the motion-referring language removed and all other settings unchanged. Neutralisation reduces camera drift substantially for Yume and Astra, leaves minWM and WorldPlay essentially unchanged, and increases drift for WorldCam. We use the neutral prompts for Yume and Astra and retain the released defaults elsewhere, following the rule “use the released default unless it explicitly contradicts the command”. The stationary results in Appendix H use these settings.

Authored interface extensions. Our additions operate only through the models’ released conditioning interfaces. We construct backward and backward-diagonal pose sequences for Astra, backward–yaw combinations for Matrix-Game diagonals, an eight-direction trajectory generator for WorldCam, and simultaneous translation–yaw commands through WorldPlay’s JSON interface. For the stationary evaluation we further supply each model with a no-op command through its native interface: identity pose sequences for the four camera-conditioned models (minWM, WorldPlay, Astra, WorldCam), an all-zero keyboard and mouse-control stream for Matrix-Game, and the composed still-clause caption for Yume. In each case the no-op is the zero element of the model’s own conditioning representation, and for the pose-conditioned models it coincides with the identity convention those releases already apply to their conditioning spans.

These inputs are representable by the released conditioning tensors but are not part of the models’ published benchmark protocols, which expose no named or evaluated no-op command; we therefore report the resulting rollouts as each model’s response to an authored zero-input extension. No model architecture, learned weight, or conditioning representation is modified.

PCA centres the tracked-motion vectors before projection, so the mean raw PCA score is zero by construction. However, the numerical score origin corresponds to the training-set mean tracked-motion field rather than to zero displacement. We therefore obtain our stationary command by passing an all-zero displacement field through the same action encoder, giving the calibrated value $a_{\text{null}} = (-0.0234, -0.0013)$ defined in Eq. (10). This correction cancels the mildly forward/right mean motion and is used only for the no-op eval-

uation; all training actions and all active commands retain their original encoded values. No analogous calibration is applied to the baselines, whose stationary inputs are the untuned zero or identity elements of their released interfaces.

D.3 Response-Curve Protocol

Ablation curves. The reverse-throttle and steering branches are measured using held-out ride windows whose recorded commands are replaced by their rotated counterfactuals. We use evaluations from the final 1,000 training steps, restrict the window offset to at most 81, and measure only settled chunks by discarding the first two generated chunks of each rollout. This is because there is a finite acceleration in the data and rapid direction changes induce a penalty from this acceleration. We find this in a lot of the other baseline models too, so all models use this protocol.

For each action axis, requested values are divided into 13 linearly spaced bins spanning the 2nd to the 98th percentile of the requested-command distribution. A bin is plotted only when it contains at least 12 samples. Each plotted point is the mean realized CoTracker-PCA response within that bin. Shaded regions show one standard error of the mean:

$$\text{SEM} = \frac{\text{std}(x)}{\sqrt{n}}.$$

Genuine reverse travel is too rare to produce a sufficiently populated rotated forward branch. We therefore measure forward response using held-out no-op contexts and the fixed positive throttle strengths

$$\{0.1, 0.2, \dots, 0.8\}.$$

Settled response is again measured after discarding the first two generated chunks. When more than 132 samples are available at a command strength, we subsample 132 examples using a fixed random seed of 0. This cap matches the size of the largest FLIP bin.

The forward and reverse branches therefore probe the same throttle coordinate but do not use identical starting-context distributions. They are used to evaluate response sign, smoothness, saturation, and approximate gain rather than as a paired forward-backward comparison. Dashed plot boundaries mark the approximate support of the training actions, extending to approximately +0.5 for forward throttle and -0.3 for reverse throttle.

Cross-model curves. Cross-model response curves use the same 32 held-out contexts as the eight-direction action-forcing benchmark. The compass commands produce the five signed per-axis protocol levels

$$\left\{-0.5, -\frac{0.5}{\sqrt{2}}, 0, \frac{0.5}{\sqrt{2}}, 0.5\right\} \approx \{-0.5, -0.354, 0, 0.354, 0.5\}.$$

For each level, the plotted point is the mean response over the 32 scenes and the shaded region shows ± 1 SEM.

Each baseline receives the corresponding fixed-strength command through its native interface. The horizontal coordinate represents the signed protocol level rather than a common physical control magnitude. The curves are therefore interpreted through response sign and within-model forward-backward asymmetry rather than absolute cross-model gain. The dotted diagonal denotes unit gain.

E Extended Controllability Results

E.1 Conditioning-Pathway Ablation

Figure 9 isolates the effect of the action-conditioning pathway. Injecting actions through both AdaLN and action tokens produces the fastest convergence and strongest counterfactual response. AdaLN alone still establishes control, but converges more slowly and reaches a weaker response. Action tokens alone appear competitive under GT evaluation but fail under FLIP, showing that continuation of the source motion can be mistaken for command following.

E.2 Training-Time Command Following

Default model. Figure 10 extends the main-paper training analysis to all eight source-motion groups for the default pca8 model with global batch size 32.

Tokens-only conditioning. Figure 11 illustrates why GT agreement is insufficient on its own. The no-AdaLN, tokens-only variant obtains high GT agreement in several common source-motion contexts, but its FLIP response remains negative or unstructured. It therefore continues motion inherited from the context without acquiring reliable command control.

Larger-batch model. Figure 12 shows the corresponding training curves for the Batch64 model. GT agreement again rises rapidly, but FLIP agreement also becomes consistently positive across all eight directions. Relative to the default batch size, the larger batch produces smoother training dynamics and a stronger response in the harder throttle directions.

E.3 Batch-Size and Critic-Supervision Ablations

Reducing either global batch size (Fig. 13) or the number of PCA components supervised by the critic (Fig. 14) delays convergence and weakens the realized reverse response, but does not prevent recovery of the backward direction.

E.4 Settled Ablation Responses

Figures 15 and 16 compare the settled directional responses of the ablation variants. Batch size has its largest effect on reverse throttle: Batch16 is substantially weaker under backward and backward-diagonal commands than Batch32 and Batch64. Steering is less sensitive to batch size and critic-supervision dimensionality, with every functional variant retaining the expected signed four-lobe structure.

E.5 Cross-Model Response Curves

Figure 17 provides a scalar view of the behavior summarized by the main-paper directional wedges. The curve construction and uncertainty estimates are defined in Section D.3.

Our model supports continuous-valued actions, whereas several baselines expose only fixed pose increments, binary controls, or text commands without a numerical strength parameter; see Section D.2, Table 8, and Table 9. Baselines are therefore shown only at strengths supported by their released interfaces, and unsupported intermediate values are not interpolated.

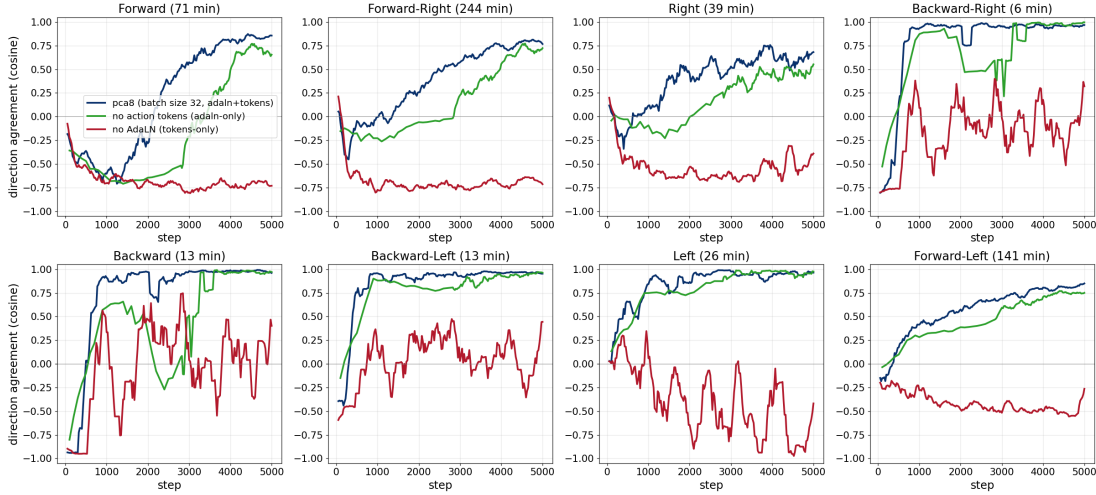


Figure 9: Conditioning-pathway ablation under counterfactual FLIP evaluation, grouped by source-ride motion direction. Dual injection through AdaLN and action tokens converges fastest and reaches the strongest response. AdaLN-only conditioning learns control more slowly, while tokens-only conditioning remains anti-aligned on throttle and unstable on steering. Panels and smoothing follow Fig. 2.

E.6 Cross-Model Steering and Yaw-Roll Coupling

Figure 18 shows that every evaluated model recovers the requested steering sign, although response magnitude and consistency across scenes differ. Figure 19 measures the roll-like component g_6 , which is not directly commanded but responds systematically to yaw across several models.

A non-zero g_6 response is not necessarily a failure. Real egocentric turns are rarely pure yaw rotations: vehicle dynamics, suspension, camera mounting, and body motion can introduce a correlated roll component. Because the PCA basis is recovered from real video, g_6 may capture this naturally coupled motion. We therefore use it as a diagnostic of yaw-roll coupling rather than assuming that all roll should vanish. Large or inconsistent responses remain undesirable, particularly when they exceed the coupling observed in real motion.

F Reference-Free Quality Evaluation Methodology

Action-forced rollouts do not admit a unique ground-truth continuation, making paired reference metrics unsuitable. We instead evaluate complementary failure modes that can be measured without a target video: visual style shift, geometric corruption, scene relocation, conjuration (spurious object appearance), and progressive loss of high-frequency detail.

Family-balanced reference. Only HF and relocation use other systems’ outputs. Under HF-Cal-v2 and Reloc-Cal-v2, the reference contains the six external baselines and one representative of our family. Default occupies that seat when an external is scored; each ablation replaces Default when scored. Relocation excludes the candidate endpoint and also uses real context references, so each candidate is matched against six other-family systems.

HF uses all seven entries, including the candidate. No variant receives same-family peer support; Default has the same score in the cross-model and ablation tables. The reference values and scoring rule are fixed, so adding an ablation does not change existing scores. These are peer-referenced diagnostics, not ground-truth comparisons.

Released reference. The files `hf_reference_dblur_v2_directional.csv` (256 inputs) and `hf_reference_dblur_v2_stationary.csv` (32 contexts) contain per-input d_{blur} values for the six externals and Default. Forming the median under the rule above reproduces the stored HF scores exactly (maximum absolute error 0.0000, no threshold-flag differences). Relocation instead requires the ORB keypoints and descriptors in `reloc_reference_pack_3000.npz`, which also stores descriptors for four real context frames. `Reloc-Cal-v2.json` identifies the eligible members and verification settings. These artifacts, rather than model weights, define the reference.

F.1 Evaluation Population

The complete action-forcing benchmark contains $32 \times 8 = 256$ directional rollouts per model. Two source contexts are unsuitable for feature-dependent evaluation because water droplets and lens smearing leave their reference frames nearly featureless. ORB detects only 371 and 104 keypoints in these contexts, compared with approximately 2,900–3,000 in representative clean contexts. Reliable ORB-RANSAC correspondence counts are therefore undefined for rollouts seeded from either context.

We exclude these two contexts uniformly from feature-dependent analyses. This removes $2 \times 8 = 16$ rollouts from every model, leaving a common 240-rollout feature-valid population. The exclusion is a property of the shared source

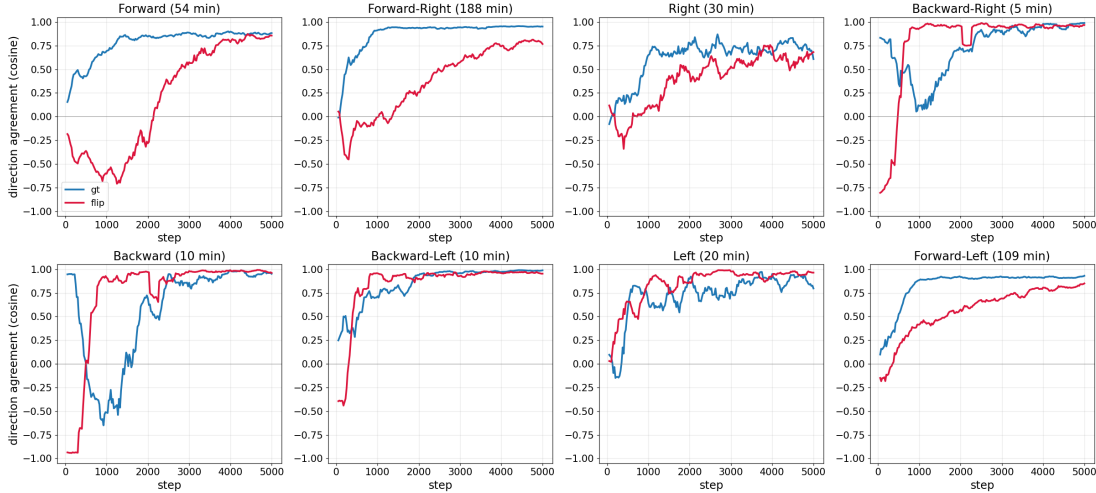


Figure 10: Training-time command following for the default model across all eight source-motion groups. Blue uses the recorded command (GT), while red applies the sign-flipped counterfactual (FLIP) to the same context. Positive FLIP agreement indicates genuine command response. Panel titles denote the source-context motion, and curves use a 9-point (≈ 450 -step) centred rolling mean.

footage rather than a model failure, and exactly the same contexts and commands are removed for every model.

A model may avoid visible generation failures by producing little or no motion. Within the feature-valid population, we therefore identify near-static rollouts using the ORB matching and homography-verification pipeline defined in Section F.4. A rollout is classified as static when its first generated frame and six-second-horizon frame retain more than 600 RANSAC-verified feature matches.

Style shift, scene relocation, and the deployed high-frequency-degradation rate are reported on the model-specific active population, defined as feature-valid and non-static. Geometry and conjunction do not require a feature-rich reference and are therefore reported over all 256 directional rollouts. The resulting feature-valid and active populations are reported in Table 10.

F.2 Geometric Corruption Probe

We detect geometric corruption—melted, dissolving, or reality-breaking structure—with a single binary probe implemented with Qwen3-VL-8B [38]. The probe receives one real reference frame followed by 16 generated frames distributed evenly over the available continuation, allowing it to assess changes across the rollout rather than at a single time point.

The reference is frame 8 of the shared real context. Generated-frame selection is deterministic: the first and final available frames within a six-second window are included, with the remaining frames placed uniformly between them. The first generated-frame indices are 9 for ours, 4 for Astra, 1 for Matrix-Game, 13 for minWM, 65 for WorldCam, and 1 for WorldPlay and Yume as they are all designed to take different amounts of context. For our 108-frame rollouts, the selected indices are $\{9, 15, 22, 28, 35, 41, 47, 54, 60, 67, 73, 79, 86, 92, 99, 105\}$.

Model or variant	Feature-valid	Active
WorldPlay	240	238
Matrix-Game	240	240
WorldCam	240	216
Astra	240	215
Yume	240	237
minWM	240	187
Ours (Default)	240	239
Ours (pca4)	240	237
Ours (pca2)	240	232
Ours (Batch 64)	240	237
Ours (Batch 16)	240	236
Ours (No Action Tokens)	240	229
Ours (No AdaLN)	240	200
Ours (No Critic)	240	75

Table 10: Quality-evaluation populations. The complete directional fleet contains 256 rollouts per model. Two wet-lens source contexts, corresponding to 16 rollouts per model, are excluded uniformly because reference-based feature matching is undefined, leaving 240 feature-valid rollouts. Active further excludes near-static continuations using the ORB-based criterion. Style, relocation, and deployed HF rates use the Active denominator.

Shorter baseline rollouts are sampled over their available duration. All images are resized to 640×352 and visibly labelled as REFERENCE or GENERATED.

The probe begins with the following preamble:

The first image labelled REFERENCE is the last real frame of a driving video—the true scene. The remaining images labelled GENERATED are an AI world model’s continuation of that exact scene, in temporal order. The model was supposed to continue the SAME scene in the SAME visual

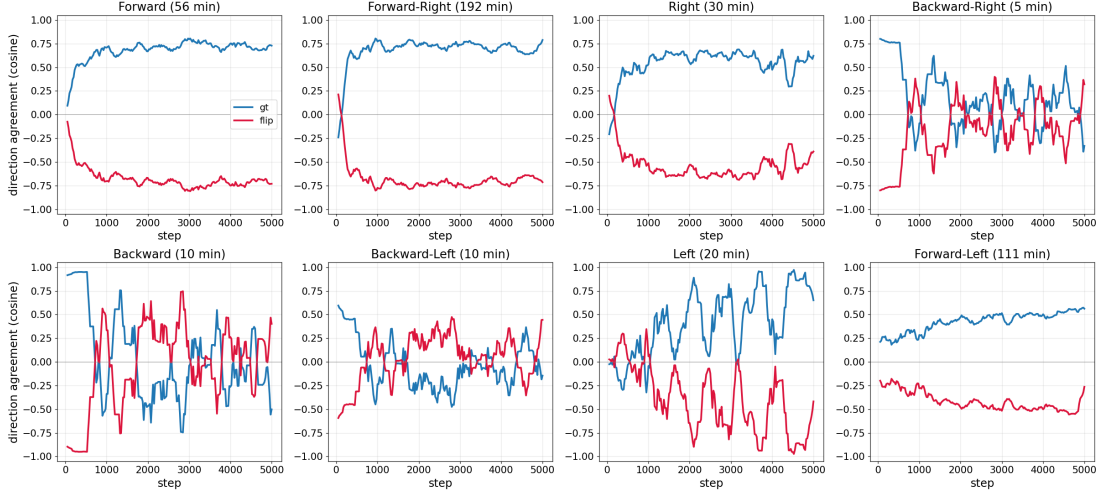


Figure 11: GT and FLIP agreement for the no-AdaLN, tokens-only variant. High GT agreement but negative or unstructured FLIP response shows that the model continues source motion without acquiring reliable command control. Panel titles denote source-context motion.

style with plausible content.

followed by the geometric-corruption question:

Is there a SIGNIFICANT uncanny or reality-breaking failure in the generated frames—impossible geometry, surfaces dissolving into abstract patterns, large corrupted regions—that a casual viewer would notice within one second? Ignore small local artifacts and minor blur. Answer Yes or No.

The question ends with the literal instruction Answer with ONLY {"answer": "Yes"} or {"answer": "No"}. The assistant response is force-prefixed with {"answer": ".". No free-form text is decoded. The probe score is the softmax probability of *Yes* against *No* at the next-token position, and a rollout is flagged when $P(\text{Yes}) > 0.5$.

F.3 Style-Shift Instrument

Style shift is a change in rendering rather than in content, so we measure it as drift in a self-supervised feature space that responds to texture and rendering statistics rather than to scene layout. We embed frames with DINOv2 ViT-S/14 [39]; each frame is resized to 224×224 and normalized with the standard ImageNet mean and standard deviation.

Two windows are compared. The anchor window is the first $\min(16, \text{ctx})$ real context frames; the end window is the last 16 generated frames within the six-second horizon. Each frame is embedded, the embeddings are ℓ_2 -normalized, and the normalized embeddings are averaged over the window, giving a single vector per window, $\bar{e}_{\text{anchor}}$ and $\bar{e}_{\text{end}}$. The style-drift score is

$$\text{dino_drift} = 1 - \cos(\bar{e}_{\text{anchor}}, \bar{e}_{\text{end}}),$$

so that a continuation re-rendered in a different visual register moves far in embedding space and scores high even when the geometry it depicts is unchanged. A rollout is flagged when its DINOv2 drift exceeds 0.72. This fixed high-drift threshold

is descriptive and is not tuned on the labelled evaluation set. Although DINOv2 does not itself require keypoint matching, we apply the common feature-valid mask because severe wet-lens corruption also makes embedding drift relative to the source frame uninterpretable.

We also evaluated a VLM style probe as a candidate—the geometric-corruption probe of Section F.2 re-asked under the identical protocol, querying whether the rendering style departs from the reference (becoming painted, game-like, cartoonish, oversaturated, or otherwise re-rendered), with ordinary lighting and exposure changes excluded. It is scored comparatively and further explored in Section G.2.

F.4 Scene-Relocation Instrument

Scene identity is measured geometrically and using feature matching instead of with VLMs. Frames are resized to 640×352 , converted to grayscale, and processed using

```
cv2.ORB_create(3000).
```

Descriptors are matched using

```
BFMatcher(NORM_HAMMING, crossCheck=True).
```

Matching therefore uses brute-force Hamming distance with mutual cross-check; no Lowe ratio test is applied.

Matches are geometrically verified using a single homography:

```
cv2.findHomography(src, dst, cv2.RANSAC, 5.0),
```

where 5.0 is the reprojection threshold in pixels. The verified-inlier count is the sum of the returned RANSAC mask.

For each source scene, Reloc-Cal-v2 supplies a frozen reference comprising one representative per family: our flagship `pca8`, the six external baselines, and the four real context references used by the released descriptor pack. The relocation score is the maximum number of RANSAC-verified

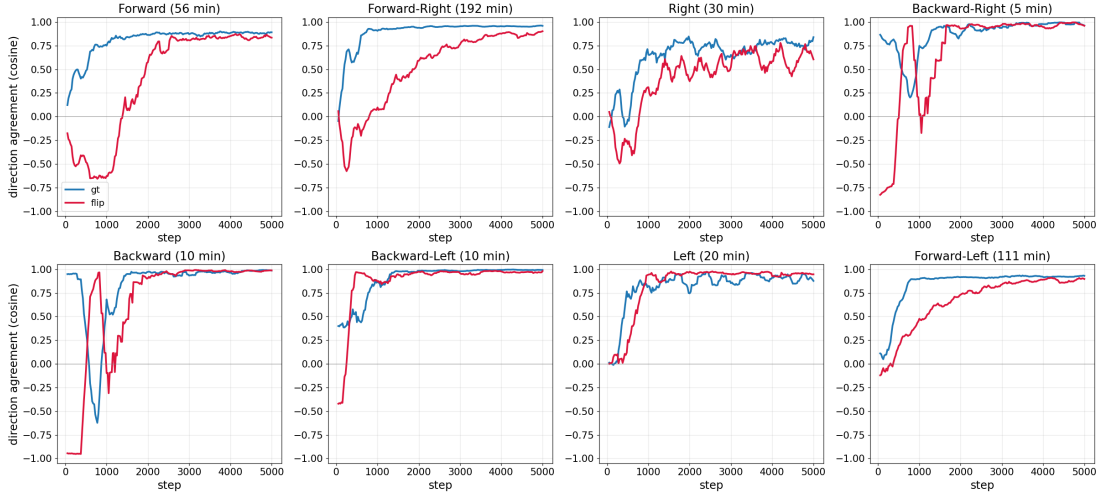


Figure 12: GT and FLIP agreement for the Batch64 model. FLIP becomes positive across all eight directions, demonstrating genuine command response. Panel titles denote source-context motion.

ORB matches to any eligible reference, and a rollout is classified as relocated when this best match contains fewer than 50 verified correspondences.

The quality-population static filter uses the same ORB, descriptor-matching, and homography pipeline. It compares each rollout’s first generated frame with its six-second-horizon end frame and classifies the rollout as static when the verified-inlier count exceeds 600.

F.5 Conjunction Instrument

A strange by-product of training on third-person video games is that there will always be an actor in the foreground who is controlled via commands, with the camera view rotating around them. Sometimes these models create such an actor even when the scene does not have one, and we label these as videos with conjured objects present, as they seem to apparate as if by magic.

In real world footage new objects appear constantly and legitimately: they enter from the side of the frame, they approach from the distance until they cross the detector’s resolution limit, and they emerge from behind occluders. Reporting any object new to the frame is therefore useless, as it flags every rollout. Requiring in addition that the object be confidently detected, born inside the central portion of the frame rather than at its border, and still present at the end removes most of these, but only reaches 0.84 AUC at 0.59 precision. Because conjunction is uncommon, low specificity is especially punishing: with a base rate of a few percent, even a small false-positive rate dominates the flagged set.

Objects are detected in every frame and linked into tracks. A candidate must begin after the context boundary, remain away from the left and right borders during its first half second, reach the confidence and minimum-area criteria, and persist near the rollout’s end. Three temporal tests then distinguish abrupt appearance from previously visible content.

The *onset* is the pixel change at the birth box divided by the change over the whole frame, so that the egomotion which

moves every pixel is divided out and only a relative spike counts; the ratio must exceed 5.0.

The *zoom probe* crops where the object is about to be, in the frames before it exists, upscales and re-detects: a distant real car is a few pixels at native resolution and looks like absence, but resolves into a confident vehicle when magnified, whereas empty ground stays empty at any magnification. A re-detection confidence of 0.50 or above counts as the object having already been there.

The *back-match* correlates the object’s own settled patch into pre-birth frames at the box its trajectory extrapolates to, catching partial occlusion reveals; a correlation of 0.65 or above likewise counts as prior appearance.

Detections come from RT-DETR R50 [40], run over all classes at a low confidence threshold so that the weak-detection tail is available to the prior-appearance tests, with tracks linked from detections scoring at least 0.30; only classes that can plausibly be conjured are reportable (car, truck, bus, motorcycle, and a small number of rigid stationary items). Each test is expressed as a signed margin from its threshold, and a rollout is flagged when the smallest of the three is positive, so all three must be satisfied independently.

The conjunction decision rule is $\text{score} > 0$. Only 13/3328 fleet rollouts lie within 0.10 of this boundary.

F.6 High-Frequency Degradation Instrument

Some variants progressively lose contrast and fine image structure during long rollouts. We measure this effect using the change in Laplacian variance, a standard indicator of high-frequency image content.

For a temporal window W , let

$$L_r(W) = \frac{1}{|W|} \sum_{t \in W} \text{Var}(\nabla^2 g_{r,t}),$$

where $g_{r,t}$ is the grayscale frame from rollout r at time t . The base window contains four frames beginning near $t = 1$ s,

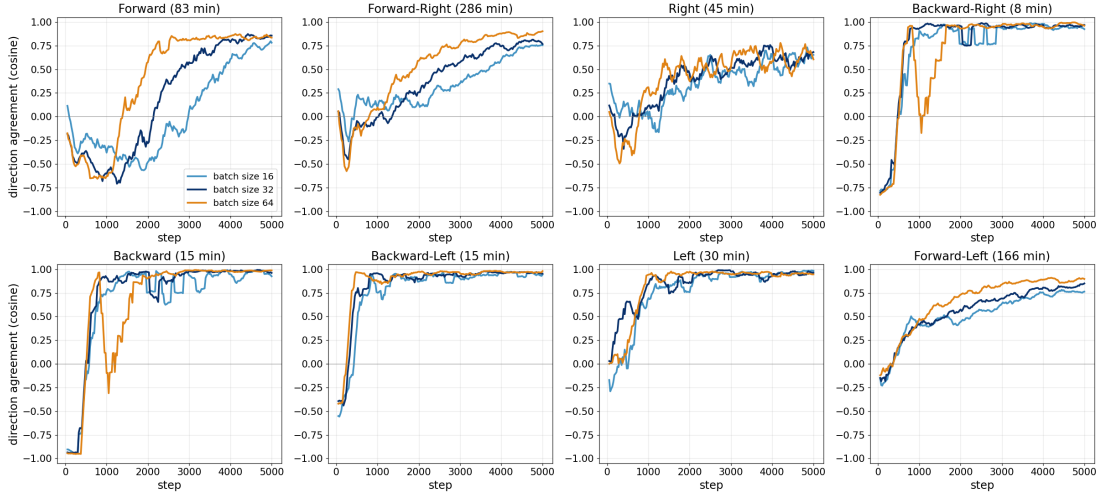


Figure 13: Batch-size ablation using global batch sizes 16, 32, and 64 under FLIP evaluation. All batch sizes recover the same directional structure by step 5,000, but larger batches acquire the hardest throttle responses earlier and retain stronger reverse motion.

while the end window contains eight frames sampled with stride two near the end of the rollout.

The loss of high-frequency detail in rollout r is

$$\Delta_r = L_r(W_{\text{base}}) - L_r(W_{\text{end}}),$$

so that larger positive values indicate a greater reduction in sharpness.

To account for scene difficulty and source-image quality, HF-Cal-v2 blocks by source scene and expresses each loss relative to the frozen one-vote-per-family reference, the six external baselines plus a single representative of our family:

$$\mathcal{B}_r = \Delta_r - \text{median}_{s \in \mathcal{S}_r} \Delta_s.$$

Positive $\mathcal{B}_r$ means that the rollout loses more high-frequency detail than the reference median for that scene, while negative values mean better relative retention. Thus an ‘‘HF 16%’’ entry means that 16% of active rollouts exceed a *peer-relative* threshold, not that only 16% visibly degrade in an absolute sense.

On 112 ablation rollouts (16 scenes, 13 haze positives), the family-balanced score achieves AUC 0.68. To summarize the continuous score as a failure rate, we classify a rollout as high-frequency degraded when $\mathcal{B}_r > 150$. This fixed reporting threshold was not optimized on the labelled evaluation set. The resulting percentage is therefore a descriptive summary of the score distribution, while the AUC evaluates the continuous metric independently of any particular threshold. Dark-channel and contrast measurements are reported only as alternative candidate metrics and do not contribute to this classification.

Score distributions. Table 11 reports median scores and threshold exceedances under HF-Cal-v2. Figures 20 and 21 show the corresponding ablation and cross-model distributions.

G Reference-Free Quality Evaluation Validation

We validate all five quality instruments against human annotations. Style shift, geometric corruption, scene relocation, and conjuration are validated on shared or enriched labelled sets. High-frequency degradation is validated separately on the ablation set, where positive examples are more concentrated, and then evaluated on the active cross-model populations defined in Section F.6.

It should be noted that evaluating these rollouts qualitatively without a reference requires nuance and subtlety; which means that some rollouts will inevitably be wrongly evaluated, however this is an unavoidable consequence of such an evaluation process, and we leave further performance gains and ensembling approaches.

G.1 Geometric Corruption

Egomotion models re-render scene geometry as the view-point changes, and their characteristic failure is localized over-warping: a surface keeps plausible colour and coarse shape while its internal structure melts, duplicates, or becomes impossible. Whole-frame pixel and line-segment statistics miss this, so we compared a VLM probe that judges melted or geometrically impossible structure against a generative artifact segmenter (PAL4VST [41]) and a melt-specific VLM prompt evaluated on left/right crops. Each candidate was scored against 36 human ‘‘mangle’’ labels on the 100-rollout subset (Table 12).

The VLM geometric-corruption probe agrees best with human judgement at 0.78 AUC. PAL4VST is close (0.77) but redundant with it. We evaluated the same probe across three open VLMs to confirm the choice of backbone: Qwen3-VL-8B is strongest, ahead of InternVL3-8B and Cosmos-Reason1-7B (Table 12). We therefore deploy the single Qwen3-VL probe and flag a rollout when $P(\text{Yes}) > 0.5$;

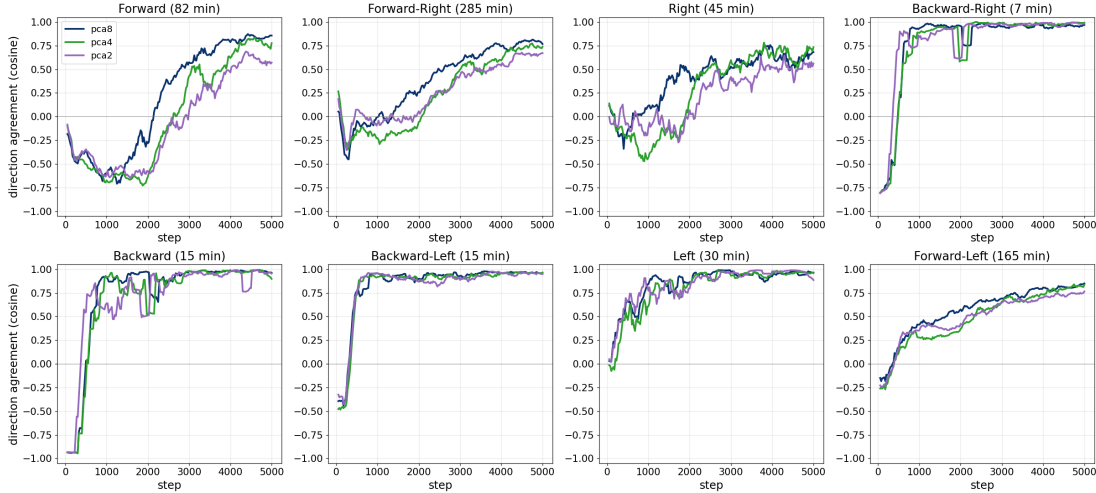


Figure 14: Critic-supervision ablation using the leading 2, 4, or 8 PCA components. Two supervised components suffice to recover directional control, while richer supervision improves convergence and realized response strength.

Variant	Median $\mathcal{B}$	$\mathcal{B} > 150$ (%)	n/N
Default	-78	16	39/239
Batch64	-51	14	33/237
pca4	-10	22	52/237
No Action Tokens	+48	29	66/229
pca2	+61	28	66/232
Batch16	+125	44	104/236
No AdaLN	+154	53	106/200
No Critic	+40	23	17/75

Table 11: High-frequency degradation under HF-Ca1-v2. $\mathcal{B}$ measures sharpness loss relative to the same-scene median of the six external baselines and the scored variant. Rates use model-specific active populations; n/N counts rollouts with $\mathcal{B} > 150$. This is a peer-relative exceedance rate, not absolute degradation prevalence. No additional sharpness-quantile exclusion is applied.

combining it with the segmenter improves AUC only within noise.

As shown in Figures 22 and 23, the geometry / warping metric earns its keep but it classifies one of the scene relocated rollouts (yume_r27) as clean, we note that this is because the metric does not find any geometry warping and it is not designed to identify scene relocation. This further necessitates separate axes controls.

G.2 Style Shifts

A style shift is a change in rendering rather than in content: the continuation departs from the reference by becoming painted, game-like, cartoonish, oversaturated, or otherwise re-rendered in a different visual register, while the underlying scene layout may stay the same.

We compared a DINOv2 embedding drift against a CLIP embedding distance, a direct style probe evaluated across three open VLMs (Qwen3-VL-8B [38], InternVL3-8B [42], and Cosmos-Reason1-7B [3]), and a classical VGG Gram-matrix style distance [43], scoring each against 20 human style labels on the 100-rollout subset (Table 13). DINOv2 drift agrees best with human judgement at 0.72 AUC: the

Detector	AUC
VLM geometric-corruption probe (Qwen3-VL-8B)	0.78
same probe, InternVL3-8B	0.75
same probe, Cosmos-Reason1-7B	0.74
PAL4VST artifact fraction	0.77
VLM melt probe, Cosmos-Reason1-7B	0.71
VLM melt probe, Qwen2.5-VL-7B	0.68
Depth-field curvature (Depth-Anything-V2)	0.52

Table 12: Geometric-corruption detectors scored against 36 human mangle labels on the 100-rollout subset. The deployed VLM probe (Qwen3-VL-8B) agrees best with human judgement; the same probe is evaluated across three open VLMs. The depth-curvature signal, part of an earlier ensemble, performs near chance (0.52) and is not used.

VLM probes asked directly whether the rendering changed land lower and disagree with one another (0.55–0.67), the CLIP distance is close but weaker (0.70), and the Gram-matrix distance is near chance. We therefore deploy DINOv2 drift.

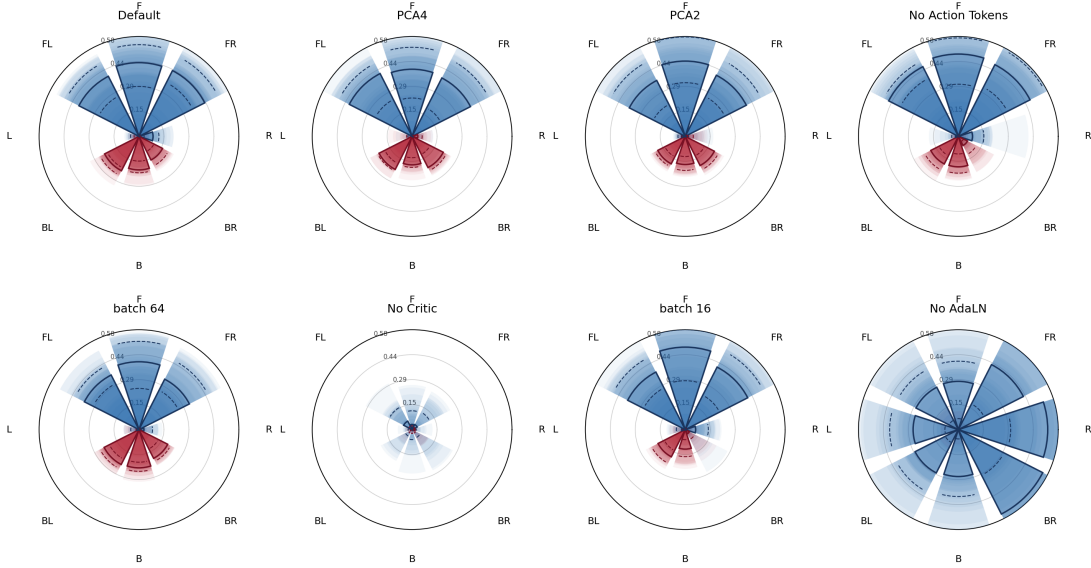


Figure 15: Settled throttle response g_0 by commanded direction for the ablation variants. Each translucent wedge represents one of 256 rollouts; blue denotes realised forward motion and red realised backward motion. Solid outlines show medians, and panel rims show the local 95th percentile. The tokens-only model renders every command as forward motion.

Signal	AUC
DINOv2 embedding drift (deployed)	0.72
CLIP embedding distance	0.70
Cosmos-Reason1-7B style probe	0.67
Qwen3-VL-8B style probe	0.66
InternVL3-8B style probe	0.55
VGG Gram-matrix distance	0.55

Table 13: Candidate style-shift measures evaluated by AUC on the human-labelled set containing 20 positives among 100 examples.

As Figures 24 and 25 show, changes in weather or lighting are classified as style shifts which is permissible depending on the definition of style. However, it should be noted that sudden weather changes are possible and plausible, but it takes nuance to judge whether the model is arbitrarily changing the weather or if it is done in a correct way, so we permit it here.

G.3 Scene Relocation

A model can abandon the input scene and generate a plausible but different place. We detect this geometrically rather than semantically: ORB keypoints are matched between the generated six-second frame and the real reference, as well as the end frames of sibling rollouts. A projective homography is fitted with RANSAC, and the number of geometrically consistent inliers is counted. A continuation that still depicts the same place retains many verified correspondences, whereas a relocated continuation does not. A rollout is flagged when fewer than 50 geometrically verified inliers survive against

Detector	AUC
ORB + RANSAC inliers (deployed)	0.70
DINOv2 embedding drift	0.72
AKAZE + RANSAC inliers	0.63
SIFT + RANSAC inliers	0.62
ORB matches, no RANSAC	0.54

Table 14: Scene-relocation detectors evaluated on a 100-video human-labelled subset containing 19 relocation positives. Higher AUC is better. ORB+RANSAC is deployed because it provides an absolute inlier count and also supports static-rollout detection. The deployed relocation decision uses this inlier count against the frozen peer panel and is therefore panel-relative rather than a standalone absolute rate.

every available reference.

We compared feature detectors and embedding-drift alternatives against 19 human relocation labels in a 100-video subset (Table 14). Among classical detectors, ORB with RANSAC agrees best with human judgement (0.70 AUC), ahead of AKAZE (0.63) and SIFT (0.62). RANSAC verification is important, as raw ORB matching drops to 0.54. DINOv2 embedding drift scores marginally higher (0.72), but provides only a relative, uncalibrated distance. We deploy ORB+RANSAC because its verified-inlier count is directly interpretable and requires no learned model, and because the same primitive supports static-rollout detection.

Representative successes and failure cases are shown in Figures 26 and 27. Figure 27 includes ambiguous flagged cases caused by post-collision content and severe lens oc-

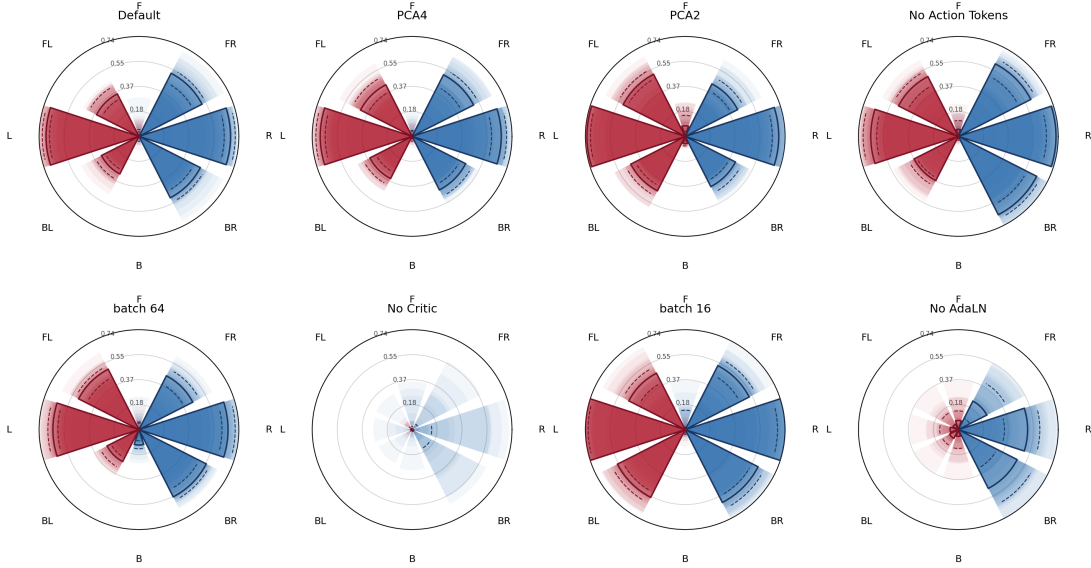


Figure 16: Settled steering response g_1 by commanded direction for the ablation variants. Blue denotes realised rightward yaw and red realised leftward yaw; opacity, outlines, and panel normalization otherwise follow Fig. 15. Every functional variant shows the expected four-lobe structure.

clusion. Both can suppress geometric correspondences without necessarily indicating true scene relocation. We retain these rare edge cases rather than tune the detector specifically around them.

G.4 Conjuring

This is rare in our clips and uncommon in most other base-lines video, but certain models are prolific in it, so rather than reuse the 100-rollout subset (which contains too few positives to separate detectors) we assemble a larger set enriched with true positives: 15 confirmed conjurations drawn from several models together with 22 rollouts confirmed clean, so that both precision and recall are measurable.

The temporal logic is agnostic to the detector backbone, so we validate that choice across three architecturally distinct backbones under an identical pipeline (Table 15). RT-DETR and Faster R-CNN [44] are indistinguishable on this set—identical precision, recall and F_1 , differing on no rollout—so we deploy RT-DETR because it is $1.7\times$ faster. RetinaNet [45] loses one further rollout, and does so only because that rollout falls exactly on the decision threshold rather than because it ranks it differently.

Figure 28 shows the instrument at work: flagged rollouts (top) have empty ground for the full second before the birth frame, after which a crisp vehicle occupies it and persists; the clean rollouts (bottom) are the hard cases a novel-object detector would fail on—a scooter present throughout, a vehicle already in the scene, and a distant vehicle approaching—each annotated with the test that rejected it.

Deployed over the full directional fleet, capped at six seconds of generation per rollout, the instrument flags 2.1% of rollouts. The rate is highly non-uniform: minWM is flagged in 19.9% of its rollouts, compared with 0.84% across our

Backbone	Prec.	Rec.	F_1	fps
RT-DETR R50 (deployed)	1.00	0.80	0.89	94.8
Faster R-CNN R50-FPNv2	1.00	0.80	0.89	56.3
RetinaNet R50-FPNv2	1.00	0.73	0.85	69.6

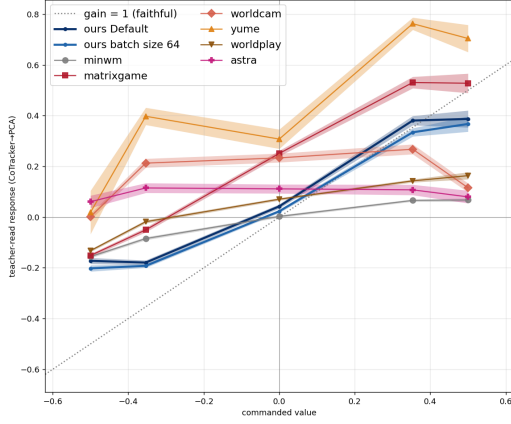
Table 15: Detection backbones under an identical temporal pipeline, scored on the enriched evaluation set: 15 confirmed conjurations drawn from several models together with 22 rollouts confirmed clean, so that both precision and recall are measurable. Throughput is end-to-end detection plus temporal analysis on a single RTX 5090.

seven variants and 0.23% across the other five external base-lines. Two controls indicate this reflects the model rather than the instrument. First, the same scenes as real footage are flagged 0 times out of 32, so the detector does not fire on genuine driving video. Second, and more telling, the gap is already present *before* any prior-appearance evidence is applied: minWM produces a persistent, crisp, newborn interior object in 52.7% of its rollouts, against 6.6–16.4% for every other model. The instrument is not simply more sensitive on minWM; minWM spawns far more new persistent objects, and a sixth of its rollouts contain one that no prior-appearance test can account for.

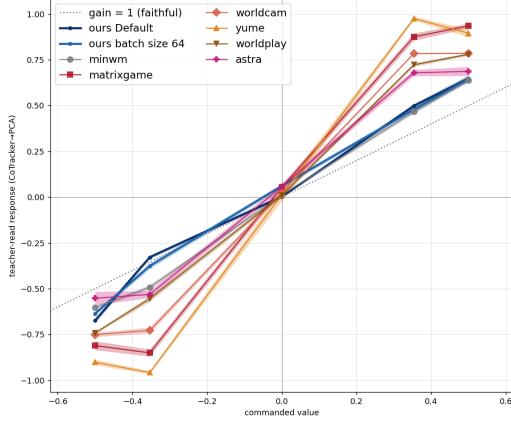
G.5 High-Frequency Degradation

High-frequency degradation is most pronounced in some of our ablations, so we validate the instrument on the 16 ablation scenes containing all seven variants (112 rollouts), of which 13 are human-labelled as degraded.

We compared the drift in Laplacian variance (a standard



(a) Throttle response.



(b) Steering response.

Figure 17: Cross-model action-forcing response curves from 32 low-egomotion held-out contexts. Realized motion is measured using the common CoTracker-PCA readout, and shaded bands show ± 1 SEM. Baselines are shown only at strengths supported by their released interfaces. The dotted diagonal denotes unit gain as a visual reference.

sharpness measure), the drift in a dark-channel haze estimate, and the drift in global contrast, each scored against those 13 haze labels (Table 16).

Figure 29 shows representative rollouts to which the high-frequency-degradation instrument is applied.

H Full Stationary Evaluation

H.1 Controllability

We apply a stationary command to every model using the protocol in Appendix D.2. We use raw PCA projections of the mean per-frame CoTracker grid flow (pixels/frame), before per-component tanh squashing. Let $p_s \in \mathbb{R}^2$ contain the first two mean raw projections for scene s , and let p_s^{real} be the corresponding real-continuation readout. We report

$$\text{Movement} = \frac{1}{32} \sum_{s=1}^{32} \|p_s - p_s^{\text{real}}\|_2.$$

Movement is thus measured in raw PCA-projection units, not pixel displacement or squashed action units; there is no additional scale factor or temporal accumulation. Lower is better. REAL is 0.0 by construction because its readout is compared with itself, not because its measured motion is exactly zero. This differs from the squashed readout used for the no-op control threshold in Section 4.3.

Figure 29 shows the corresponding departures.

Animation Low Movement alone can reward a frozen scene. Animation therefore measures residual CoTracker motion after removing the top three PCA modes. Tracks are binned into an 8×8 grid; a cell is alive if it has at least three tracks, residual magnitude above 2 pixels and coherence ratio $\|\text{mean}\|/\text{mean}(\text{std}) > 1.0$. Animation is the mean count of alive cells (0–64). It is a proxy for independent scene activity, not an object count: one mover can activate several cells.

Two confounds remain. Depth-dependent parallax can inflate Animation in strongly drifting models. Native bitrate also differs: our variants encode at about 7.5 Mbps and external outputs at about 1.6 Mbps; Animation has a weak positive association with bitrate (Spearman 0.185, $p = 0.003$, $n = 256$). Matrix-Game’s and WorldPlay’s near-zero values therefore suggest suppressed scene activity but are not an unconfounded measure of freezing. Table 17 reports both axes.

H.2 Quality

Table 18 reports stationary quality failure rates across the model variants and external baselines. Our full-capacity variants have 0% style shift, 0% geometric corruption and 3% relocation. Matrix-Game and Astra show geometric corruption in 94% and 44% of stationary rollouts, respectively. Peer-referenced relocation is 41% for Astra, 25% for Yume and 16% for WorldCam. These failures are distinct from the no-op motion test: low measured camera motion need not imply an intact scene.

The one ablation that stands out is `noadaln`, which washes out in 66% of stationary rollouts (median $\mathcal{B} = +228$ under HF-Cal-v2) and relocates in 12%.

I Training Infrastructure and Statistical Protocol

I.1 Hardware and software

All models were trained on an HPE Cray EX system with NVIDIA GH200 Grace-Hopper nodes: four GPUs per node, 95 GiB of usable HBM per GPU, compute capability 9.0, 128 CPU cores per task, and a Slingshot-11 interconnect (NCCL over the `hsn` interfaces). Training used PyTorch 2.10.0 with CUDA 12.8, distributed data-parallel across nodes, and LoRA adapters over a frozen Wan2.1-T2V-1.3B backbone.

The Default model and every conditioning ablation (PCA-2, PCA-4, No-Act-Tokens, No-AdaLN, No-Critic) were trained on 8 nodes = 32 GPUs at global batch 32 and learning rate 4.95×10^{-5} . The batch-size ablations used 4 nodes = 16 GPUs (global batch 16, lr 3.5×10^{-5}) and 16 nodes = 64 GPUs (global batch 64, lr 7.0×10^{-5}). Every variant trains

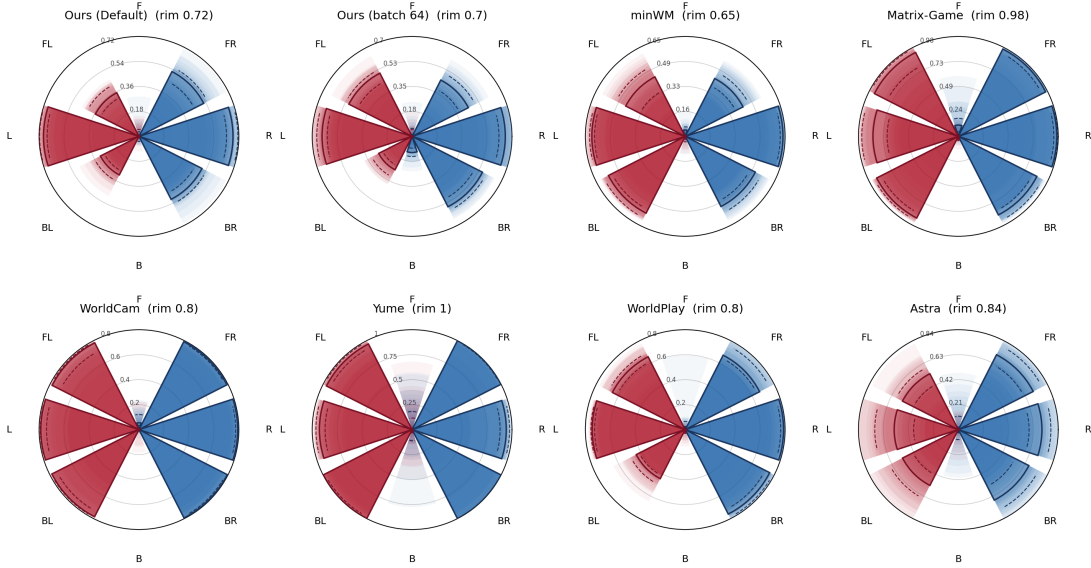


Figure 18: Steering response g_1 by commanded direction for all models. Blue denotes realised rightward yaw and red realised leftward yaw; other wedge conventions follow the main-paper throttle figure. Every model recovers the expected steering sign, although response magnitude and variance differ.

Detector	AUC
Laplacian drift, original analysis	0.85
Laplacian drift, original-panel rescore	0.78
Laplacian drift, family-balanced (deployed)	0.68
Dark-channel haze drift, sibling-relative	0.82
Dark-channel haze drift, raw	0.77
Laplacian-variance drift, raw	0.68
Contrast drift	0.64

Table 16: High-frequency-degradation validation on 112 rollouts with 13 haze positives. The first three rows separate the original Laplacian analysis from rescoring under the original and family-balanced references. Other candidate results are from the original comparison, not a new matched comparison against $v2$.

from step 0 to step 5000. The scheduler enforces a four-hour limit per job, so each run proceeds as a short sequence of checkpoint-resumed segments; Table 19 reports the total.

I.2 One run per variant, many rollouts per run

Each reported model is a **single training run** at `seed: 0`, offset per rank so that data order and initialisation are deterministic. We report no seed replicates. At roughly 850 GPU-hours per variant, a three-seed replication of the eight-model table would cost on the order of 22,000 GPU-hours, which was not available to us; we therefore spent the budget on breadth of evaluation instead of depth of replication.

Training is not bit-reproducible past the first step, because the backward pass uses non-deterministic kernels. Measured run-to-run loss agreement is $\sim 10^{-3}$, against a loss that moves by ~ 0.09 between adjacent steps of a single run, so the residual is well inside one run’s own variation.

Generation itself is also hardware-sensitive along the autoregressive chain. With identical weights and seeds on dif-

ferent GPU hardware, first chunks are near-identical (mean absolute difference in frame mean 0.43) but diverge by a factor of roughly 30 by chunk 8 (13.08). The stationary Movement statistic is robust in the measured rerun (per-scene $r = 0.995$, maximum $|\Delta| = 2.2$), whereas Animation is not ($r = 0.50$; for example $r07\ 9 \rightarrow 0$ and $r17\ 19 \rightarrow 1$). Stationary-table reruns should therefore use generations from the same hardware. Hardware robustness was not measured for HF, relocation, geometry, conjuration, or style, and we make no stability claim for those five metrics.

I.3 Evaluation Uncertainty

Each of our variants is trained once. Each evaluated model produces 256 directional and 32 no-op rollouts. Legitimacy is computed on the common feature-valid subset of 240 directional rollouts: 30 source contexts with eight commands each. These rollouts measure evaluation variation, not independent training runs.

For the Default–minWM legitimacy comparison, we ac-

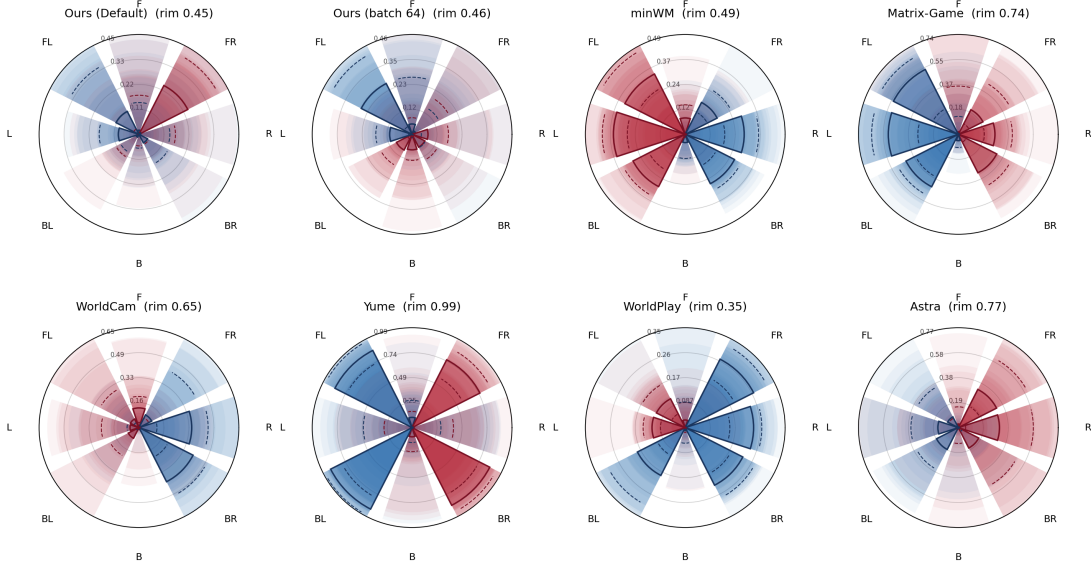


Figure 19: Uncommanded roll response g_6 under each planar command. Radius indicates roll magnitude; blue denotes clockwise and red counter-clockwise motion. Direction-dependent responses reveal coupling between commanded planar motion and out-of-plane camera motion.

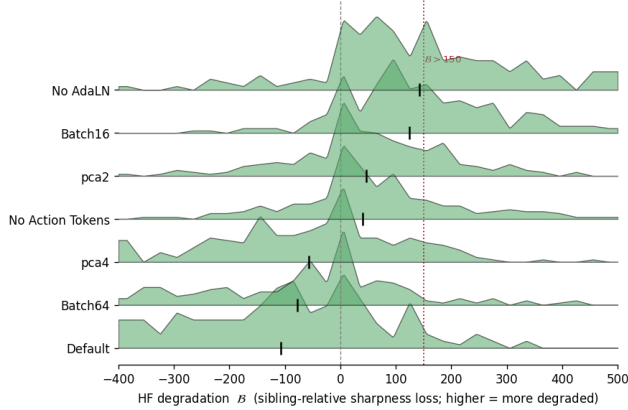


Figure 20: Distribution of the high-frequency-degradation score B under HF-Cal-v2 (reference-relative sharpness loss; higher is more degraded). No AdaLN and Batch16 degrade most, while Default, Batch64, and pca4 lie at or below zero and preserve detail better than the same-scene reference continuations.

count for shared source contexts by keeping all eight commands and cross-model pairing together. We bootstrap the 30 contexts with replacement 20,000 times and take the 2.5th and 97.5th percentiles of the mean paired difference as a 95% confidence interval. A two-sided paired t -test compares the models' per-context legitimacy rates.

Bootstrap resampling uses `numpy.random.default_rng(0)`. The analysis is reproduced by running `grids/eval/legit_`

`cluster_inference.py` end-to-end.

Default achieves 55.0% legitimacy (132/240), versus 44.2% for minWM (106/240). The observed difference is 10.8 percentage points, with a context-bootstrap 95% interval of $[-4.6, +26.2]$ points and paired-test $p = 0.19$. We therefore report an empirical benchmark advantage without claiming statistical significance. Other model comparisons are descriptive unless explicitly tested. We wish to highlight that we train on unsupervised video with a recovered affine action basis whereas minwm need well labelled data.

This uncertainty analysis conditions on the trained checkpoints and fixed peer reference; it does not quantify training-seed or reference-panel variation. Response-curve SEM bands are unadjusted rollout-level summaries, while wedge arcs show interquartile ranges across rollouts. Neither represents variation across training runs.

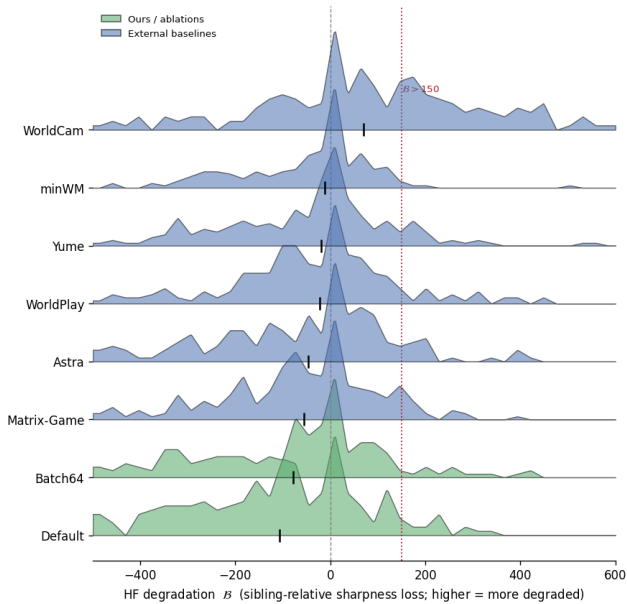


Figure 21: Cross-model distribution of the high-frequency-degradation score $\mathcal{B}$ under HF-Cal-v2 for the six external baselines, Default and Batch64, using their active evaluation populations (1,809 rollouts in total). Higher values indicate greater reference-relative loss of sharpness. Green denotes our models, while blue denotes external baselines. Black ticks mark medians, the grey dashed line marks $\mathcal{B} = 0$, and the red dotted line marks the descriptive threshold $\mathcal{B} > 150$.

Model	Movement	Animation
REAL (reference)	0.0	1.1
Ours (Default)	2.3	1.4
Ours (pca4)	2.1	0.0
Ours (pca2)	2.2	0.0
Ours (Batch64)	2.0	0.6
Ours (Batch16)	2.0	0.0
Ours (No Action Tokens)	2.1	0.1
Ours (No AdaLN)	32.2	6.2
Ours (No Critic)	1.8	0.2
minWM	3.0	0.2
Matrix-Game	2.1	0.0
WorldPlay	2.3	0.0
WorldCam	6.3	2.8
Astra	19.8	11.2
Yume	45.9	0.8

Table 17: Stationarity and animation over 32 null-command contexts. *Movement* is the mean distance between generated and real-continuation readouts in raw PCA-projection units, before tanh; lower is better and REAL = 0.0 by construction. *Animation* is the mean count of alive cells in an 8×8 grid after removing the top three PCA motion modes, not a count of objects. A faithful no-op should hold the viewpoint without freezing the world. Animation is most interpretable for low-Movement models and is partly confounded by native bitrate.

Model	Style %	Geom. %	Reloc. %	Conj. %	HF %
Default	0	0	3	0	3
pca4	0	0	3	0	0
pca2	0	0	3	0	9
Batch64	0	0	3	0	0
Batch16	0	0	3	0	12
No Act. Tok.	0	0	3	0	3
No AdaLN	0	3	12	0	66
No Critic	0	0	3	0	6
minWM	0	0	6	31	9
Matrix-Game	0	94	9	0	3
WorldPlay	3	16	12	0	12
WorldCam	3	19	16	0	38
Astra	6	44	41	0	6
Yume	0	6	25	0	16

Table 18: Stationary-rollout failure rates (%); lower is better. Style, Geom., Reloc., Conj., and HF denote style shift, geometric corruption, scene relocation, conjuration, and high-frequency degradation. HF and relocation are peer-referenced against the family-balanced v2 reference (one seat per family; a variant of ours occupies our seat itself when scored); the remaining quality axes are panel-independent.

Variant	Nodes	GPUs	Wall-clock	GPU-h
Default (PCA-8)	8	32	26.9 h	862
Batch64	16	64	28.4 h	1819
Batch16	4	16	27.4 h	439
PCA-4	8	32	27.5 h	879
PCA-2	8	32	26.9 h	860
No-Act-Tokens	8	32	21.0 h	671
No-AdaLN	8	32	26.7 h	854
No-Critic	8	32	26.8 h	857
Total				$\approx 7,250$

Table 19: Measured training cost per reported model.

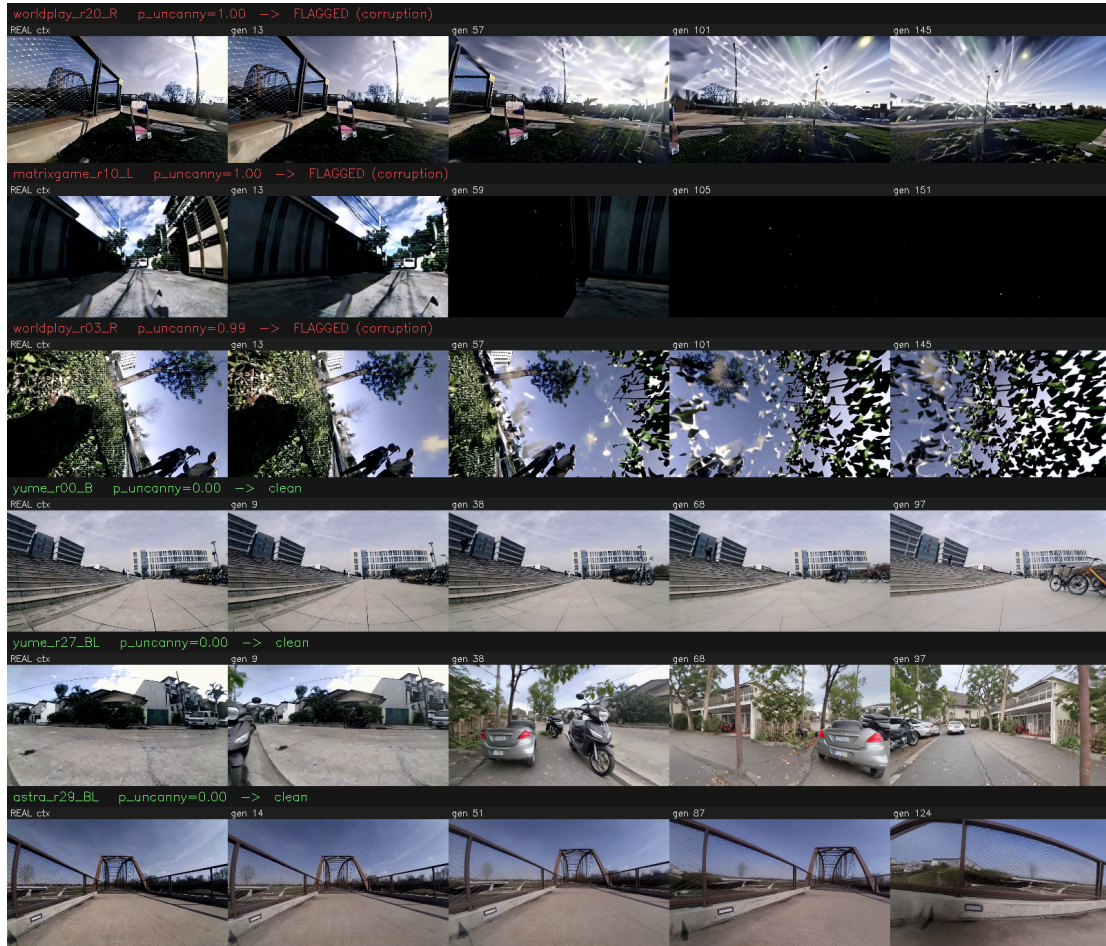


Figure 22: Geometry-instrument examples for the external baselines, showing representative human-positive and human-negative cases. Yume r27 relocates without visible geometric corruption, illustrating why relocation needs a separate instrument.



Figure 23: Geometry-instrument examples for our ablation variants, showing representative human-positive and human-negative cases.



Figure 24: Style-shift-instrument examples for the external baselines, showing representative flagged and clean rollouts.



Figure 25: Style-shift-instrument examples for our ablation variants, showing representative flagged and clean rollouts.



Figure 26: Scene-relocation-instrument examples for the external baselines, showing representative relocated and retained-scene rollouts.



Figure 27: Scene-relocation-instrument examples for our ablation variants. Post-collision content and severe lens occlusion illustrate ambiguous flagged cases that suppress geometric correspondences without necessarily changing the underlying scene.



Figure 28: Conjunction-instrument examples. Top: flagged rollouts centred on object birth, showing empty ground beforehand and a persistent object afterwards. Bottom: hard negatives in which new objects arrive legitimately; annotations identify the prior-appearance test that rejected each candidate.



The high-frequency-degradation instrument on our ablation variants. High- B rollouts (top) visibly wash out over the generated span—texture and contrast are lost by the end—while low- B rollouts (bottom) preserve detail. The same source scenes appear in both classes, so the effect is model-dependent, not scene-dependent.

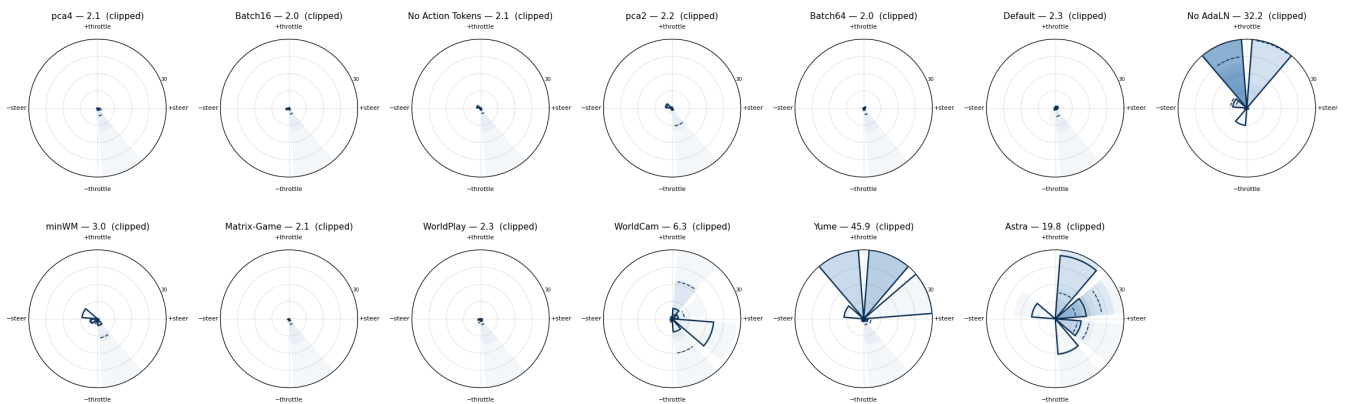


Figure 29: Residual egomotion under the null command. Each translucent wedge represents one of 32 contexts. Its angle and radius describe the two-component raw PCA departure from the real continuation; radius uses the same raw PCA-projection units as Movement, before tanh squashing. Blue denotes forward drift and red backward drift. Solid outlines show medians and dashed arcs show interquartile ranges.