ROCKETWM: A FRAMEWORK FOR BUILDING CONTROLLABLE WORLD MODELS FOR AEROSPACE FLIGHT

Anonymous authors
Paper under double-blind review

ABSTRACT

Controllable World Models (WMs) of aerospace flight are currently only feasible for those with proprietary data: flight logs that pair video with commands belong to the launch provider, which leaves competitors' vehicles, retired vehicles, and historical launches without a simulator. Public launch broadcasts are openly available but show only what happened, and their on-screen telemetry describes the vehicle's state rather than its commands. A video WM trained on such footage therefore learns appearance and trajectory statistics but not the effect of control commands: its action channel improves reconstruction during training, yet generation remains unresponsive to actions at inference. Methods that turn observation-only video into action-conditioned training data are thus lacking, a gap this work aims to fill. We present RocketWM, a method that generates physics-grounded controls from launch vehicle footage alone. We extract the observed flight dynamics from the on-screen telemetry, initialise an aerospace physics engine with them, and render the resulting trajectories in a game engine whose camera is matched to the broadcast. The resulting physics twin reproduces the recorded flight within the telemetry's own tolerance. Varying the twin's commands yields paired counterfactual episodes with rendered ground-truth video, and mapping its best-fitting commands back onto the real footage produces an action-conditioned launch dataset. We fine-tune two WMs on this dataset to predict several seconds ahead and compare them against conventional telemetry-based conditioning on two flight phases, booster and ship landing burns. The resulting WMs respond to control commands, and their generated trajectories closely follow the real footage. Although demonstrated on aerospace launch vehicles, our framework is designed to be applicable in any domain where video and state are observed but commands are not.

1 INTRODUCTION

World models describe how the world evolves and how an agent's actions change future observations (Ha & Schmidhuber, 2018; NVIDIA et al., 2026a). Generative video world models have become simulators for robots and cars because those domains record both halves of that sentence, observations paired with the actions that caused them (Yang et al., 2024; Bruce et al., 2024; Russell et al., 2025; NVIDIA et al., 2026b). The pairing is what allows the model be driven under commands that were never executed.

A world model of rocket flight would be valuable for the same reason, and more so. An aerospace vehicle cannot be flown a thousand times to rehearse a landing burn, and the logs that pair its video with commands exist only for the provider's own vehicle, and only once it has flown. Public footage is the only record of other programmes' vehicles, of retired ones, and of a new vehicle before its logs accumulate; a simulator built from it is also a starting point that a provider's own logs can later refine. The public record of a flight, however, is broadcast footage whose on-screen telemetry shows speed, altitude, which engines are lit and an attitude indicator: the state, shown late and coarsely, and never the throttle, thrust direction or mass that produced it. Conditioning a video model on this telemetry looks like action conditioning, and by the usual measure it works, because reconstruction of the recorded future improves. We show that the improvement is not control: the generated video does not change when the telemetry is removed, and a change of random seed moves it 7.6 times more than deleting every action (Section 2). Two properties of launch data explain why. A rocket flies one trajectory per flight, so the mission clock predicts every telemetry channel and nothing forces the

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

model to depend on the channel rather than on the flight it recognises. In addition, the commands the footage does reveal, ignitions and cut-offs, are so rare that a model trained to reconstruct the future never needs them.

Our idea is to obtain the commands from physics. The display shows which engines are lit, but not throttle, thrust direction or mass. The equations of motion supply the rest: the path fixes the specific force that produced it, and with the lit engines known, thrust direction, mass and throttle follow algebraically. We build a *physics twin* of each flight on this principle, a six-degree-of-freedom model whose commands are recovered by inversion, not search (Section 3). From the twin we take two things the footage lacks. Every real frame receives the commands in force at that instant, on the vehicle's own clock rather than the display's delayed one. In principle, the twin can be flown again from any recorded state under a changed command and rendered with cameras matched to the broadcast, which gives *counterfactual pairs*, two flights from one state that differ in a single command, so the outcome that did not happen exists as video (Section 4). Rendered clips enter training while held-out pairs form CommandBench, our custom benchmark.

We call the framework RocketWM (Section 5) and make three contributions. First, we show that telemetry conditioning improves reconstruction without giving control, that is, without the generated video changing when the command changes, and identify the two properties of launch data responsible (Section 2). Second, we propose command recovery by inversion through an explicit model of the broadcast display; a reconstruction is accepted only if its re-flight matches the broadcast and the same test refuses two deliberately corrupted command histories (Section 3). Third, we build an action-conditioned aerospace dataset in which every label carries its value, its uncertainty and its source, with held-out counterfactual pairs, and post-train two video world models on it (Sections 4–6).[1] On the upper stage's landing burn, the model trained on footage and renders follows the commanded ignition on 56 of 64 held-out pairs while the same model trained on footage alone follows on none. In addition, the booster's landing burn shows the same pattern. All results concern one aerospace vehicle family.

2 Problem setting: footage shows states and hides commands

Footage. A broadcast is a video $o_{1:T}$ with a telemetry display. Behind it a vehicle with state $\boldsymbol{q}(t)$ (position, velocity, attitude, mass) is driven by commands $\boldsymbol{u}(t)$: engines lit, throttle, thrust direction, surface deflections. Following NVIDIA et al. (2026a) we call *actions* the causal variables that change the state; here they are the commands, and the broadcast never shows them. It shows

$$y_j^{\mathrm{obs}}(t) \;=\; Q_{\Delta}\big[\,(g_\kappa * y_j)(t-\delta_j)\,\big], \qquad y_j = h_j(\boldsymbol{q}), \tag{1}$$

a function of the state through the *display model* of channel j, a low-pass filter (time constant κ_j), a delay δ_j of one to five seconds, and a quantiser of step Δ_j (0.1 km and 1 km/h on recent flights; Appendix B.1).

A state channel cannot be shown to steer. A video world model predicts $p_\theta(o_{P+1:T} \mid o_{1:P}, \boldsymbol{a}_{1:T})$; launch footage offers as $\boldsymbol{a}_t$ only the telemetry, a function of the state. Let $\boldsymbol{c}$ be the context available without the conditioning sequence: conditioning frames, mission clock, flight phase, camera view. Write channel k as $a_k = \psi_k(\boldsymbol{c}) + \varepsilon_k$, with ψ_k its best predictor from $\boldsymbol{c}$.

Definition 1 (Command index)**.** $\iota_k = 1 - R^2(a_k \mid \boldsymbol{c})$*, estimated with the flight held out.*

Proposition 1 (No overlap, no constraint)**.** *Let $g^\star(\boldsymbol{c}, \boldsymbol{a})$ minimise a training risk $\mathbb{E}\,\ell(g(\boldsymbol{c}, \boldsymbol{a}), o_{P+1:T})$. If $\iota_k = 0$ for every k, every g that agrees with $g^\star$ on the graph $\{(\boldsymbol{c}, \psi(\boldsymbol{c}))\}$ attains the same risk, including $\tilde{g}(\boldsymbol{c}, \boldsymbol{a}) = g^\star(\boldsymbol{c}, \psi(\boldsymbol{c}))$, which ignores $\boldsymbol{a}$. The response to $\boldsymbol{a}' \neq \psi(\boldsymbol{c})$ is not constrained by training.*

The proof is immediate (Appendix A.1); the content is that the premise holds. With the flight held out, context predicts the upper stage's altitude and speed at $R^2 = 0.99$, and none of the 24 telemetry channels is a command. The proposition also says why such a channel *helps training*: on the graph, $\boldsymbol{a}$

[1] Closest to this work: counterfactual video from a language-derived scene description, never rendered independently (Shen et al., 2025); action-conditioned drone video from recorded inertial actions (Radi et al., 2026); physics identified from video of tabletop objects (Li et al., 2023; Jiang et al., 2025). None identifies a vehicle from its own real footage or tests a model against a rendered counterfactual (Section 7).

summarises the future and lowers the loss, which reconstruction metrics reward, while conditioning on a state selects which recorded video to reproduce and does not intervene on the cause (Ahuja et al., 2023). We measured this on the secondary backbone (Section 5). Post-trained on 12 flights, the telemetry-conditioned model gains 2.05 dB PSNR over the same model trained without the telemetry vector; at inference, deleting the vector changes PSNR against the true future by 0.003 dB, and changing the seed moves the output 7.6 times more than deleting every action, with a same-command control at chance (Appendix G).

Commands are rare inside a window. The commands the footage does reveal, changes in the number of lit engines, are sparse. Model their times as a stationary Cox process with mean rate λ; let the *coverage* $C(W)$ be the probability that a uniformly placed window of length W contains one.

Proposition 2 (Coverage under bursty commands)**.** $C(W) = 1 - \mathbb{E}[e^{-\Lambda_W}] \le 1 - e^{-\lambda W}$*, with* Λ_W *the random intensity integrated over the window, and equality if and only if the process is Poisson.*

On our corpus one command occurs per 166 s. A 3.1 s window contains one in 0.88 % of uniform draws against a Poisson distribution 1.79 % at 121 s coverage saturates at 10.5 % (Appendix C.4).

Requirement. Training data must contain (i) channels with positive command index; (ii) at a matched context, more than one value of a command with its visual consequence; (iii) windows placed where commands change. Footage supplies none. We recover (i) by physics, manufacture (ii) with the same physics engine, and obtain (iii) by sampling.

3 THE PHYSICS TWIN: RECOVERING COMMANDS BY INVERSION

The *physics twin* of a flight is a six-degree-of-freedom flight-dynamics model whose state follows the broadcast telemetry, so that its commands are the commands that flew. The usual construction is a search over a forward simulation, yet we show why that fails on this data and replace it with a method that contains no search (Figure 1, top row).

3.1 WHY SEARCHING OVER A FORWARD SIMULATION FAILS

Keep only thrust T along a body axis $\hat{\boldsymbol{b}}$ and gravity: $m\,\ddot{\boldsymbol{p}} = T\hat{\boldsymbol{b}} + m\boldsymbol{g}$, $\dot{m} = -T/c_{\text{eff}}$.

Proposition 3 (Thrust–mass scale symmetry)**.** *For every* $\lambda > 0$ *the histories* $(\lambda m, \lambda T)$ *fly the same path* $\boldsymbol{p}(t)$*. From the path alone only the thrust per unit mass* $s = T/m$ *is identifiable; the mass scale is not. Drag* D *breaks the symmetry only in proportion to* D/T*.*

A search over propellant load, throttle, dry mass and a drag scale therefore walks a valley of equal loss: on one flight three seeds of an evolution-strategy search ended at drag scales of 2.00, 1.40 and 1.98 at the same loss to five digits, while the same optimiser recovers a planted interior optimum on every seed (Appendix B.5). A search has a second defect: nothing ties its state to the observed state at a given instant except the loss, so a landing burn that it ignites at an altitude turns a state error into a timing error. Our search-based twin lit its landing burn 14.5 s late while with our method events flown on the *measured* schedule exact to 0.1 s.

3.2 DISPLAY MODEL AND PATH RECONSTRUCTION

Speed and altitude alone do not pin the delays of Eq. (1); the likelihood is flat to ± 0.5 s. We fix them from two physical facts: the vehicle is at rest until release, so the displayed speed ramp must extrapolate to zero at release, and the attitude indicator fixes the lag of altitude relative to speed (Appendix B.1). On a known-answer test with a planted display, reconstruction recovers s and incidence to 0.25 % and 0.43° with the display model, and to 2.00 % and 3.24° without it.

We then make the observed trajectory the state. In the vertical plane of the ground track the reconstruction state is $[\, d, h, \dot{d}, \dot{h}, f_d, f_h \,]$, downrange, altitude, their rates and the *specific force* $\boldsymbol{f}$ (the acceleration not due to gravity or the Earth's rotation), modelled as a random walk and observed through displayed speed and altitude at their delayed time stamps; a fixed-interval smoother returns the path with its covariance. One choice carries the physics: *the process noise of the specific force is*

opened only at the instants at which the engine-state indicator changes, so a thrust step is permitted exactly where the broadcast shows engines lighting or shutting down (Appendix B.3).

3.3 COMMAND RECOVERY

With speed V, flight-path angle γ and geocentric distance r, resolve the specific force along the velocity and normal to it in the vertical plane:

$$f_t = \dot{V} - (\boldsymbol{g} + \boldsymbol{c}_{\mathrm{f}})\cdot\hat{\boldsymbol{t}}, \qquad f_n = V\dot{\gamma} - V^2\cos\gamma/r - (\boldsymbol{g} + \boldsymbol{c}_{\mathrm{f}} + \boldsymbol{c}_{\mathrm{c}})\cdot\hat{\boldsymbol{n}}, \tag{2}$$

where $\boldsymbol{c}_{\mathrm{f}}$, $\boldsymbol{c}_{\mathrm{c}}$ are the centrifugal and Coriolis accelerations. In **powered flight**, with α the incidence (velocity to thrust axis), D the axial drag and N_{b} the body normal force,

$$s = \mathrm{hypot}\Big(f_t + \tfrac{D}{m},\, f_n - \tfrac{N_{\mathrm{b}}(\alpha)}{m}\Big), \qquad \alpha = \mathrm{atan2}\Big(f_n - \tfrac{N_{\mathrm{b}}(\alpha)}{m},\, f_t + \tfrac{D}{m}\Big). \tag{3}$$

Thrust per unit mass and thrust direction are *algebraic* in the path: no integration, hence no drift and no clock of their own. This is differential flatness with the path as flat output (Fliess et al., 1995). With $N(t)$ lit engines read from the engine-state indicator and the thrust law $T = N(\eta\, T_{\mathrm{vac}} - pA_e)$, mass follows by one quadrature and throttle by division:

$$\dot{m} = -(s\,m + N\,p\,A_e)/c_{\mathrm{vac}}, \qquad \eta = (s\,m + N\,p\,A_e)/(N\,T_{\mathrm{vac}}). \tag{4}$$

The scale that Proposition 3 withholds enters once, as m_0, from outside the kinematics: the displayed propellant gauge, profiled over m_0 with $\eta \le 1$ as a one-sided bound, identifies the lift-off mass to 0.54 % in a leave-one-flight-out test (Appendix B.3). In **unpowered flight at high incidence** (atmospheric entry), drag and the vertical component of lift give two equations in three unknowns, incidence, bank and a crossflow-density scale; the attitude indicator's nose angle is the third, and flap deflections follow from the moment balance at the recovered attitude.

Each label records its class: engine on/off states are *measured* (± 0.02 s), throttle and thrust direction are *algebraic*, flap deflections are identified *through trim* and not scored as commands, and reaction-control thrusters, fast gimbal motion and the sign of bank are *not recoverable* and are not labelled (Table 6).

3.4 FORWARD RE-FLIGHT AND ACCEPTANCE

The recovered commands are flown in JSBSim (Berndt, 2004) on the measured engine schedule, under a weak tracking loop and then *open loop* to the end of the burn. The re-flight is the consistency check and the state from which counterfactuals branch. A reconstruction is accepted only if the re-flight meets two criteria: (a) the robust z-score of its speed and altitude against the displayed telemetry, scaled by the reading's own spread, is at most 1.5; and (b) at the end of the open loop, altitude is within 1 %, speed within 0.5 % and flight-path angle within 1°. The same criteria must *refuse two planted faults* flown beside it, thrust aligned with the velocity and throttle lowered by 2 % (Appendix B.4).

4 AN ACTION-CONDITIONED FLIGHT DATASET

4.1 DATA PIPELINE

Figure 4 shows the pipeline from broadcast to labelled clip. We use the Starship vehicle[2] because its broadcasts carry a rich telemetry display and its flights span ascent, entry and powered landing of two stages. Broadcasts are normalised, cut at shot boundaries into 3–10 s clips, filtered for blur, exposure and static countdown, and judged by a vision–language critic. Duplicates are removed twice, by appearance (perceptual hash and Cosmos-Embed1 embedding similarity) and by identity (audio landmarks and the displayed mission clock), so that the same seconds of a flight never sit on both sides of a split. Each clip is cropped to a display-free rendition centred on the vehicle, located by Grounding DINO and SAM 2.1 (Liu et al., 2025; Ravi et al., 2025). The display is read at 2 fps (digits by PaddleOCR (Cui et al., 2025), the engine indicator by per-engine luminance, gauges by radial

[2] https://www.spacex.com/vehicles/starship

sampling, the attitude indicator by a ResNet-18 head), smoothed by a constant-acceleration Kalman filter and Rauch–Tung–Striebel Sunesh et al. (2026) pass onto a 10 Hz grid, and fused with indicator transitions, subtitles and audio into flight events. On hand-read samples our method is exact on speed, altitude and clock (100 %), and the same samples shifted by 3.0 s score 5–40 % (Appendix C). These readings are the twin's inputs. At the next step, the twin returns per flight a 10 Hz table of state and commands with a standard deviation per cell.

4.2 Labels on the vehicle's clock, with their sources

Each clip is labelled by sampling the twin at the mission time of each frame. Displayed speed and altitude trail the vehicle by two to four seconds while engine states, read from the frame, do not, so labels are placed on the pixel clock. Every engine event in the labels lies within 0.2 s of its frame-read time (204/204, median 0.017 s), and the twin's standard deviation covers its residual against the display (97.9 % and 96.7 % of rows within 2σ). Labels are layered, which means that each cell records value, σ and source in the trust order *observed* > *engine indicator* > *kinematic* > *displayed* > *trim*, and a layer may *refute* a lower cell it can show to be wrong (in practice 7.6 million cells). A registry assigns each of the 24 channels per vehicle a role, *command*, *state* or *reserved*; a command is scored only where a camera view resolves its effect. Under that rule the upper stage carries four engine commands and the booster its throttle; flaps and the upper stage's throttle are labelled but reserved (Appendix C.3). The final dataset holds 5,061 real clips (11.1 h) of 12 flights and 2,052 rendered clips (6.8 h), split 4,581/480 with one flight held out entirely. In total 104 real clips contain an engine command change.

4.3 Counterfactual pairs

A *counterfactual pair* is two flights of the forward twin from the same seeded state under command sequences that differ in one commanded quantity. The dataset has 6 sets of 120 pairs: ascent engine-out and throttle, boostback engine set and throttle, booster landing-burn delay, and the upper stage's landing-burn engine count. Our custom benchmark (6.1) adds landing-burn ignition altitude shifted by ± 120 m. Both members, and 306 replay episodes of every flight phase, are rendered in Unreal Engine 5.8.2, from viewpoints matched to the broadcast's shot types (chase, hull-mounted, an aft engine-bay view, pad tower). A set is kept only if the intervention is visible on at least one viewpoint. Replay episodes and pairs enter training as a fixed share of the draws (30 % in the mixed variant) while held-out pairs with disjoint seeds are the benchmark (Appendix D).

4.4 Sampling and mixing

Only 2.5 % of uniformly placed upper-stage windows contain an engine change (Proposition 2), so we set the mix of windows: a window class (a change, steadily on, steadily off, unlabelled) is drawn first, the change's offset inside the window is drawn uniformly, and the start is placed from it. Let d be a clip's domain (real or render), e a binary command state with $\Pr(e{=}1 \mid d) = \pi_d$, and suppose the context does not predict e within a domain.

Proposition 4 (Mixture shortcut)**.** *If the domain is decodable from the inputs, then for any render share $\rho \in (0, 1)$ the command state is predictable from the inputs with balanced accuracy above $\frac{1}{2}$ if and only if $\pi_{\mathrm{render}} \neq \pi_{\mathrm{real}}$.*

We observed this: a test predicting engine state from context scored 0.50 on real clips and on renders alone and 0.76 on their mixture. Base rates matter, not the mixing ratio, so the trainer draws the class first, then the domain at a fixed share within the class, then the clip (Appendix C.4).

5 RocketWM: a world model for aerospace flight

RocketWM is our proposed framework and the high-level overview is depicted in Figure 1. The physics twin turns a broadcast into commands, the commands label the real clips and, re-flown and rendered, supply counterfactual pairs, a video world model is post-trained on both with the vehicle treated as a robot with an initial state followed by commands and CommandBench tests whether its generations move with the command.

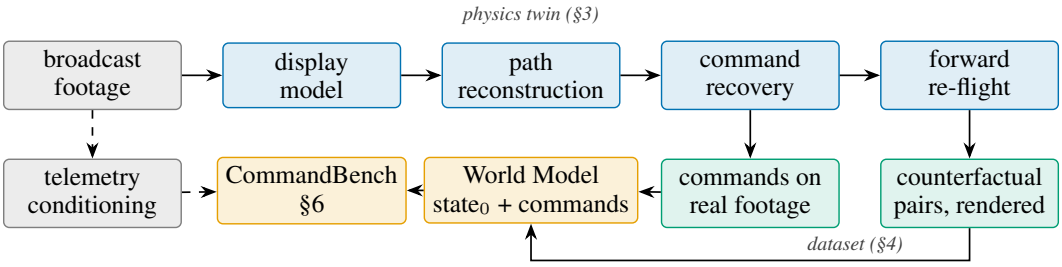


Figure 1: RocketWM. Top: the physics twin turns a broadcast into commands in one direction, reading the telemetry through a model of the display, reconstructing the path, and inverting the equations of motion; a re-flight checks the result. Middle: the twin yields commands for the real clips and rendered counterfactual pairs. Bottom: a world model conditioned on initial state and commands is tested against the render of the commanded outcome, beside the telemetry-conditioned baseline (dashed).

Backbones. *Cosmos 3 Edge* (NVIDIA et al., 2026a), a 3.4 B mixture-of-transformers model, treats action as a modality: in forward-dynamics mode a clean action token per step precedes the noisy video tokens in one sequence, and each embodiment has its own input and output projection. We add the aerospace vehicle as a new embodiment in the last free slot; its projection is re-initialised and the other embodiments' rows take no gradient. Windows are 61 frames at 10 fps (6 s, 832×480), the labels' native rate. The loss is on vision only, no guidance or dropout is applied to actions and no adapters are used. The generator tower and action interface (1.4 B parameters) are post-trained at learning rate 10^{-4} ($\times 5$ on the action modules) for 6,000 iterations (Appendix E). *Cosmos-Predict2.5-2B* (NVIDIA et al., 2026b), the secondary backbone, is a latent diffusion transformer trained by flow matching (Lipman et al., 2023) that admits the action through an embedding added to the time embedding of every block. We train rank-32 adapters on 93 frames (3.1 s). Its telemetry variant is the diagnostic of Section 2.

Conditioning. For each of up to two vehicles in view, $\boldsymbol{a}_t \in \mathbb{R}^{24}$ carries a validity mask and a partition into *command channels* (engine states) and *state channels* (altitude, speed, attitude, propellant and their rates; Appendix C.3). A masked channel selects a learned "absent" embedding, so that unknown and zero differ. The vehicle is conditioned as a robot is: in the **robot form** the state channels are valid on the first row of the window only and the command channels on every row, so the consequence of a command must be generated rather than read off a future state. Two ablations change one thing each. Under **full-window state** every valid channel is visible on every row; the **real-only** variant keeps the robot form but trains without renders. The **telemetry** baseline conditions on the full window of displayed telemetry without the twin.

6 Experiments

6.1 CommandBench

Sets. Held-out counterfactual pairs of the upper stage's landing burn, rendered on the chase rig from the accepted twin, with seeds disjoint from every training export: ignition altitude shifted by ± 120 m (*ignite_time*, $n = 64$), a late ignition (*level_on*, $n = 61$) and an early cut-off (*level_off*, $n = 4$, underpowered). In addition ten held-out pairs of the booster's landing burn with a delayed ignition, on the engine-bay and chase rigs. **Conditions per pair.** True generates from the pair's shared conditioning frames under its own commands; Swap under the other member's; Self repeats True at a second seed and must not pass; Null removes every action; State gives the other member's window state with the model's own commands (full-window variant only). **Statistic.** With Δ_g the onset shift between Swap and True and Δ_c the shift between the two renders, a pair is won when the signs agree, $|\Delta_g| \geq 0.2$ s and the gain $\Delta_g/\Delta_c \geq 0.25$. We report wins over all pairs, so ties and missing read-outs count against, with an exact Clopper–Pearson interval and Holm correction across variants, and the median gain. **Fidelity.** PSNR, SSIM and LPIPS (Zhang et al., 2018) against the real continuation and a content-debiased Fréchet video distance (Ge et al., 2024) against held-out

Table 1: Physics twin acceptance on 12 flights. Stack ascent: robust z of altitude and speed against the reading's own spread (bar 1.5), planted faults refused (throttle $-2\,\%$; thrust along the velocity); end-of-open-loop errors in Appendix B.4. Landing burns: accepted burns with their recovered throttle. Lift-off masses from the propellant gauge.

flight	m_0 (t)	z_h	z_V	faults	landing burn
IFT-2	4,843	0.82	0.03	2/2	–
IFT-3	4,964	0.83	0.09	2/2	booster: not passed
IFT-4	4,955	0.93	0.05	2/2	booster 0.70
IFT-5	4,970	0.82	0.03	1/2	booster 0.73
IFT-6	4,955	0.88	0.02	1/2	booster 0.64; ship 0.58
F7	5,331	0.79	0.05	2/2	booster 0.80
F8	5,283	0.91	0.10	2/2	booster 0.74
F9	5,098	0.13	0.06	2/2	–
F10	5,222	0.06	0.02	2/2	ship: not observed
F11	5,241	0.07	0.05	2/2	ship 0.49
F12	5,497	0.10	0.14	2/2	ship 0.77
F13	5,565	0.07	0.04	2/2	ship: not passed

Table 2: Command following on held-out landing-burn pairs, Cosmos 3 Edge, iteration 6,000: wins/losses/ties, exact 95 % interval on wins over all pairs, median gain. SELF: the same command at a second seed (wins on ignite_time; level_on).

	ignite_time ($n = 64$)			*level_on* ($n = 61$)			
Variant	W/L/T	95 % CI	gain	W/L/T	95 % CI	gain	SELF
Upper-stage landing burn, robot form							
M0 (renders)	64/0/0	[0.94, 1.00]	0.92	61/0/0	[0.94, 1.00]	0.94	5; 0
H (real + renders)	56/0/8	[0.77, 0.94]	0.64	56/0/5	[0.82, 0.97]	0.66	18; 8
H-real (real)	0/2/62	[0.00, 0.06]	0.00	0/0/61	[0.00, 0.06]	0.00	–
M0-cap (renders, captions)	64/0/0	[0.94, 1.00]	1.00	61/0/0	[0.94, 1.00]	0.97	3; 0
Ablation: full-window state							
H-full (real + renders)	44/0/20	[0.56, 0.80]	0.46	53/0/8	[0.76, 0.94]	0.67	18; 7
Booster landing burn, ignition delay, 10 held-out pairs							
	engine-bay rig			chase rig			
M0, seed 1 (renders)	10/0/0	[0.69, 1.00]	+1.09	10/0/0	[0.69, 1.00]	+0.98	2; 1
M0, seed 2 (renders)	8/0/2[b]	[0.44, 0.97]	+0.93	6/0/4[b]	[0.26, 0.88]	+0.84	2; 1
H (real + renders)	0/0/10	[0.00, 0.31]	+0.00	0/0/10	[0.00, 0.31]	+0.00	2; 0

real clips, with the real-versus-real value as its floor. **Variants.** Every Edge variant shares the recipe of Section 5 and differs in training data or conditioning form only; the real-only set is cut from the mixed one by removal. The upper-stage variants trained on the version 4.2 of the dataset (four engine commands, 1,888 renders), the booster variants on the version 5.0.

6.2 RESULTS

R1. The twin is accepted on every ascent and on most landing burns (Table 1). The stack ascent passes the two conditions on 12/12 flights and the planted throttle fault is refused on all. 25 ascent burns and 8 of 10 scored landing burns are accepted with a throttle. Two landing burns fail the condition and 3 have no engine indicator in view.

R2. Models trained on recovered commands follow them; the same model trained on real footage alone does not. Table 2. The render-only variant moves the ignition with the command on every pair. The variant trained on real footage and renders follows on 56 of 64 and 56 of 61 pairs (median gain 0.64), and the real-only variant follows on none and draws no plume on 57 of 64. SELF scores at or below chance on every variant and NULL never wins. The full-window variant follows more weakly, and its STATE swap wins 0/64; 1/61 pairs: window state does not stand in for the command.

R3. Obedience is bought with realism, and the mixture recovers it. Table 3. The render-only variant, which follows every command, is far from the broadcast; the mixed variant sits at the real-versus-real floor on the distributional distance and within 0.3 dB of the real-only variant on PSNR.

Table 3: Fidelity on 16 held-out real landing-burn clips, three seeds; cd-FVD ± seed spread, real-versus-real floor 613 ± 60 (21-clip bench: 824 ± 90; Appendix G). Predict2.5 variants were not scored

variant	PSNR ↑	SSIM ↑	LPIPS ↓	cd-FVD ↓
M0 (renders)	13.80	0.472	0.575	1,291 ± 48
H (real + renders)	18.88	0.693	0.288	617 ± 48
H-full	19.05	0.696	0.291	626 ± 63
H-real	19.18	0.701	0.283	596 ± 49
M0-cap (renders, captions)†	14.18	0.471	0.558	1,234 ± 59
real versus real	–	–	–	613 ± 60

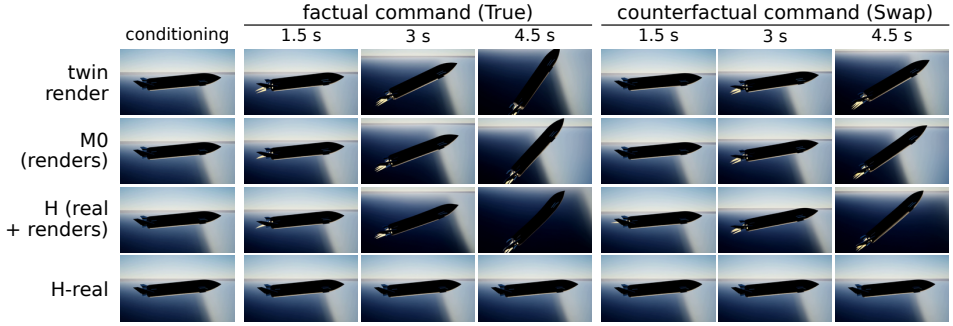


Figure 2: One held-out rendered pair: the twin's renders (top) and each variant's generation from the shared conditioning frame under the factual (TRUE) and counterfactual (SWAP) command at 1.5, 3 and 4.5 s. The plume appears at 1.5 s under TRUE and later under SWAP for the render-only and mixed variants while the real-only variant draws none. Full-window variant and a real clip: Figure 10.

R4. What carries the effect. Figure 2 and Appendix G. Adding real footage to the renders cuts the loss-probe effect about eightfold (15/15, $+0.020$; 17/19, $+0.028$ against 13/15, $+0.005$; 14/19, $+0.005$) and the video gain from 0.92 to 0.64while removing the renders clears it completely. In the render-only variants the probe effect grows with training without saturating (Figure 9) and replicates on two further seeds and on the Nano 16B backbone. In the mixed variant it stays within the probe's run-to-run spread, so there the video bench, not the probe, carries the evidence (Appendix G). The telemetry variant of the same backbone is null on the probe while its caption control is live (6/15, -0.0002 (text control 12/15)), so the instrument sees conditioning on that variant and the command carries nothing. Interestingly the Predict2.5 probe reads (8/15 [0.27, 0.79], 9/15 [0.32, 0.84]) are inconclusive, so that backbone contributes the null of Section 2 only. The embodiment row moves as far in the real-only variant as in the mixed one (7.5 % against 7.3 %) while it follows on nothing.

R5. Booster landing burn: the same pattern. Table 2, lower block. On ten held-out pairs with a delayed booster ignition, the render-only variant follows on 10/0/0 (engine-bay rig, gain +1.09) and 10/0/0 (chase rig) at one training seed and on 8/0/2 at a second; the mixed variant follows on none.

7 RELATED WORK

World models and their actions. Latent world models learn dynamics for planning and control (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2025; Hansen et al., 2024; Micheli et al., 2023; Zhou et al., 2025); generative video world models make the simulation itself the interface (Yang et al., 2024; Bruce et al., 2024; Alonso et al., 2024; Russell et al., 2025; NVIDIA et al., 2026b;a). Their actions are recorded, or are pseudo-actions from pose differences, or are latent actions learned from video alone (Schmidt & Jiang, 2024; Ye et al., 2025; Baker et al., 2022). Ours are commands that were never recorded, recovered from mechanics with units and uncertainty. Richens & Everitt (2024) show that a robust agent must have learnt a causal model of its environment; Proposition 1 is the data-side converse, since without overlap no causal model of the command is learnt.

Measuring control. Action-conditioned models are scored by reconstruction against the recorded future under the recorded actions (NVIDIA et al., 2026b; Zhu et al., 2025), which Proposition 1 shows cannot separate a model that follows actions from one that ignores them; conditioning inputs are known shortcuts in driving, where ego status alone explains open-loop planning scores (Li et al., 2024; Geirhos et al., 2020). Where adherence is measured it is by re-estimating the control from the output, with detectors on generated driving scenes (Russell et al., 2025; Gao et al., 2024; Arai et al., 2025) or an action chunk replayed in a simulator (NVIDIA et al., 2026a). CommandBench adds a rendered counterfactual reference, a same-command control that must sit at chance. Benchmarks of physical plausibility (Motamed et al., 2026; Li et al., 2026; Meng et al., 2025; Gu et al., 2026) score what a generation looks like while ours scores whether it changed with the command.

Physics from video and flight reconstruction. Differentiable simulation and rendering recover parameters of tabletop objects from multi-view video (Li et al., 2023; Jiang et al., 2025; Xie et al., 2024; Zhang et al., 2025); Wang et al. (2026) give identifiability conditions for second-order dynamics from video, of which Proposition 3 is the launch-vehicle case. Counterfactual video from a language-derived scene description has no rendered outcome (Shen et al., 2025); simulator renders have been translated to photorealistic video (NVIDIA et al., 2025) and mixed with real data (Zhou et al., 2024; Maddukuri et al., 2025), and Proposition 4 adds a condition the mixing share cannot meet. Best-estimate trajectories from flight measurements are established practice (Morelli & Klein, 2016; Maine & Iliff, 1986; Karlgaard et al., 2013). We found no precedent for reconstruction from a broadcast display, with a model of that display, used to label real video with commands. Flight control and world models for drones use recorded actions (Kaufmann et al., 2023; Radi et al., 2026), and powered-descent guidance is learned in simulation (Gaudet et al., 2020) and none is grounded in the real footage of an orbital launch vehicle.

8 LIMITATIONS

One vehicle family. Every flight belongs to one aerospace vehicle. We have not investigated transfer to another vehicle, and the display model is specific to one broadcaster's graphics. **The mass scale is external.** By Proposition 3 it cannot come from the path while ours comes from a propellant gauge and is conditional on an assumed tank capacity (Appendix B.3). **Unrecoverable commands.** Reaction-control thrusters, fast gimbal motion and the sign of the bank angle are not recovered. Flap deflections are recovered through trim and have not been validated against a view of a flap, so they are not scored as commands.**Cameras are matched, not calibrated**, and the real half of the mixture retains a context shortcut that only more real clips containing command changes can remove (Appendix C.4).**Scale.** Our post-training budget is five orders of magnitude below published action-conditioning recipes; a null at this budget is evidence about the budget (Section 5). These results come with several limitations, spanning the scope of the vehicle data, what the physics twin can and cannot recover, and the scale at which we post-train. We nonetheless conclude that the gap between reconstruction and control is real, measurable, and addressable by grounding actions in physics rather than in the broadcast footage alone.

9 CONCLUSION

Footage of a physical system records what its state did, not what it was told to do, and a world model conditioned on the former is better at reconstruction and no more controllable. Where the governing equations are known, the missing commands can be recovered from the footage itself, with their limits of identifiability stated, and the same physics supplies the counterfactuals against which control is measured. We present RocketWM, a framework for building controllable World Model for aerospace flight grounded in physics. Our result highlight that on one aerospace vehicle's landing burn, a video world model trained this way follows the commanded ignition on 56 of 64 held-out pairs where the same model trained on footage alone follows on none. Future work includes more training steps of the WMs, additional aerospace vehicle families and learning control policies using the WMs.

ETHICS STATEMENT

The dataset is derived from publicly broadcast launch footage. Upon acceptance, we will release labels, derived telemetry, twin tables, rendered clips and code, and identifiers and time ranges that locate the source footage. We do not redistribute broadcast video. The work concerns data and simulation for learning. It makes no claim about operational guidance, navigation or control of a real launch vehicle, and the physical parameters it uses are published specifications or quantities estimated from public video.

AI USE STATEMENT

AI model assistants were used in different parts of this work. 1) A custom harness was build to iteratively refine and improve the dataset quality with LLMs, VLMs and VLAs as decision makers and judges. 2) An extension of this harness was used to verify and validate all the results that are presented this article. 3) Frontier LLMs were used to polish the language and assist on drafting this article. All content was reviewed and curated by the authors.

REPRODUCIBILITY STATEMENT

Every number outside Section 6.2 is traced to a named artefact in the supplementary material; records are addressed by configuration hash, never by recency. The supplementary code contains the display model, the reconstruction and inversion with their known-answer tests, the acceptance test with its planted faults, the pair factory and its identity check, the sampler, the evaluation statistics (exact intervals, sign test, detectable effect at 80 % power), and the configuration difference of every pair of settings compared in a table. *The full codebase will be released upon acceptance*

REFERENCES

Kartik Ahuja, Divyat Mahajan, Yixin Wang, and Yoshua Bengio. Interventional causal representation learning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=NadTwTODgC.

Hidehisa Arai, Keishi Ishihara, Tsubasa Takahashi, and Yu Yamaguchi. ACT-bench: Towards action controllable world models for autonomous driving. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025. URL https://openreview.net/forum?id=26KlsDgwLi.

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24639–24654. Curran Associates, Inc., 2022. doi: 10.52202/068431-1789. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9c7008aff45b5d8f0973b23e1a22ada0-Paper-Conference.pdf.

Jon S. Berndt. JSBSim: An open source flight dynamics model in C++. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, pp. 4923, Providence, RI, USA, August 2004. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2004-4923.

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 Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and

Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, Yue Zhang, Wenyu Lv, Kui Huang, Yichao Zhang, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. Paddleocr 3.0 technical report, 2025. URL https://arxiv.org/abs/2507.05595.

Michel Fliess, Jean Lèvine, Philippe Martin, and Pierre Rouchon. Flatness and defect of nonlinear systems: introductory theory and examples. *International Journal of Control*, 61(6): 1327–1361, 1995. doi: 10.1080/00207179508921959. URL https://doi.org/10.1080/00207179508921959.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Tw9nfNyOMy.

Brian Gaudet, Richard Linares, and Roberto Furfaro. Deep reinforcement learning for six degree-of-freedom planetary landing. *Advances in Space Research*, 65(7):1723–1741, 2020. ISSN 0273-1177. doi: https://doi.org/10.1016/j.asr.2019.12.030. URL https://www.sciencedirect.com/science/article/pii/S0273117719309305.

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

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Jing Gu, Xian Liu, Yu Zeng, Ashwin Nagarajan, Fangrui Zhu, Daniel Hong, Yue Fan, Qianqi Yan, Kaiwen Zhou, Ming-Yu Liu, and Xin Eric Wang. $phyworldbench$: A comprehensive evaluation of physical realism in text-to-video models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=rlZeILv3fm.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

Nick Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 47376–47405, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/cf73d57b6dcda32b293df7c2d5341f49-Paper-Conference.pdf.

Nikolaus Hansen. The cma evolution strategy: A tutorial, 2023. URL https://arxiv.org/abs/1604.00772.

Hanxiao Jiang, Hao-Yu Hsu, Kaifeng Zhang, Hsin-Ni Yu, Shenlong Wang, and Yunzhu Li. Phystwin: Physics-informed reconstruction and simulation of deformable objects from videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7219–7230, 2025. doi: 10.1109/ICCV51701.2025.00678.

Christopher D Karlgaard, Roger E Beck, Stephen D Derry, Jay M Brandon, Brett R Starr, Paul V Tartabini, and Aaron D Olds. Ares ix trajectory reconstruction: Methodology and results. *Journal of Spacecraft and Rockets*, 50(3):641–661, 2013.

Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Champion-level drone racing using deep reinforcement learning. *Nature*, 620 (7976):982–987, 2023.

Dacheng Li, Yunhao Fang, Yukang Chen, Shuo Yang, Shiyi Cao, Justin Wong, Michael Luo, Xiaolong Wang, Hongxu Yin, Joseph E. Gonzalez, Ion Stoica, Song Han, and Yao Lu. Worldmodelbench: Judging video generation models as world models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=a3hafrDzuA.

Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chenfanfu Jiang, and Chuang Gan. PAC-neRF: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tVkrbkz42vc.

Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M. Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14864–14873, 2024. doi: 10.1109/CVPR52733.2024.01408.

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

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 38–55, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72970-6.

Abhiram Maddukuri, Zhenyu Jiang, Lawrence Yunliang Chen, Soroush Nasiriany, Yuqi Xie, Yu Fang, Wenqi Huang, Zu Wang, Zhenjia Xu, Nikita Chernyadev, Scott Reed, Ken Goldberg, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, Los Angeles, CA, USA, 2025.

Richard E Maine and Kenneth W Iliff. Application of parameter estimation to aircraft stability and control: The output-error approach. Technical report, NASA, 1986.

Fanqing Meng, Jiaqi Liao, Xinyu Tan, Quanfeng Lu, Wenqi Shao, Kaipeng Zhang, Yu Cheng, Dianqi Li, and Ping Luo. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 43781–43806. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/meng25c.html.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vhFu1Acb0xb.

Eugene A Morelli and Vladislav Klein. *Aircraft system identification: theory and practice*, volume 2. Sunflyte Enterprises Williamsburg, VA, 2016.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 948–958, 2026. doi: 10.1109/WACV61042.2026.00099.

NVIDIA, :, Hassan Abu Alhaija, Jose Alvarez, Maciej Bala, Tiffany Cai, Tianshi Cao, Liz Cha, Joshua Chen, Mike Chen, Francesco Ferroni, Sanja Fidler, Dieter Fox, Yunhao Ge, Jinwei Gu, Ali Hassani, Michael Isaev, Pooya Jannaty, Shiyi Lan, Tobias Lasser, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Fabio Ramos, Xuanchi Ren, Tianchang Shen, Xinglong Sun, Shitao Tang, Ting-Chun Wang, Jay Wu, Jiashu Xu, Stella Xu, Kevin Xie, Yuchong Ye, Xiaodong Yang, Xiaohui Zeng, and Yu Zeng. Cosmos-transfer1: Conditional world generation with adaptive multimodal control, 2025. URL https://arxiv.org/abs/2503.14492.

NVIDIA, :, Aditi, Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, Aarti Basant, Mukesh Beladiya, Mohammad Qazim Bhat, Zaid Pervaiz Bhat, Dan Blick, Vanni Brighella, Han Cai, Tiffany Cai, Eric Cameracci, Jiaxin Cao, Yulong Cao, Mark Carlson, Carlos Casanova, Ting-Yun Chang, Yan Chang, Yu-Wei Chao, Prithvijit Chattopadhyay, Roshan Chaudhari, Chieh-Yun Chen, Junyu Chen, Ke Chen, Qizhi Chen, Wenkai Chen, Xiaotong Chen, Yu Chen, An-Chieh Cheng, Click Cheng, Xiu Chia, Jeana Choi, Chaeyeon Chung, Wenyan Cong, Yin Cui, Magdalena Dadela, Nalin Dadhich, Wenliang Dai, Joyjit Daw, Alperen Degirmenci, Rodrigo Vieira Del Monte, Robert Denomme, Sameer Dharur, Marco Di Lucca, Ke Ding, Wenhao Ding, Yifan Ding, Yuzhu Dong, Nicole Drumheller, Yilun Du, Aigul Dzhumamuratova, Aleksandr Efitorov, Hamid Eghbalzadeh, Naomi Eigbe, Imad El Hanafi, Hassan Eslami, Benedikt Falk, Jiaojiao Fan, Jim Fan, Amol Fasale, Sergiy Fefilatyev, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Vikram Fugro, Prashant Gaikwad, TJ Galda, Katelyn Gao, Yihuai Gao, Wenhang Ge, Sreyan Ghosh, Arushi Goel, Vivek Goel, Akash Gokul, Rama Govindaraju, Jinwei Gu, Miguel Guerrero, Elfie Guo, Aryaman Gupta, Siddharth Gururani, Hugo Hadfield, Song Han, Ankur Handa, Zekun Hao, Mohammad Harrim, Ali Hassani, Nathan Hayes-Roth, Yufan He, Chris Helvig, Cyrus Hogg, Madison Huang, Michael Huang, Sophia Huang, Yufan Huang, Jacob Huffman, DeLesley Hutchins, Suneel Indupuru, Boris Ivanovic, Arihant Jain, Joel Jang, Ryan Ji, Yanan Jian, Dongfu Jiang, Jingyi Jin, Atharva Joshi, Nikhilesh Joshi, Pranjali Joshi, Andy Ju, Jaehun Jung, Weiwei Kang, Scott Kassekert, Jan Kautz, Ashna Khetan, Julia Kiczka, Slawek Kierat, Gwanghyun Kim, Kuno Kim, Sunny Kim, Kezhi Kong, Xin Kong, Zhifeng Kong, Tomasz Kornuta, Egor Krivov, Hui Kuang, Saurav Kumar, Chia-Wen Kuo, George Kurian, Wojciech Kutak, JF Lafleche, Himangshu Lahkar, Omar Laymoun, Jayjun Lee, Sanggil Lee, Gabriele Leone, Boyi Li, Freya Li, Jiajun Li, Jinfeng Li, Ling Li, Pengcheng Li, Shangru Li, Tingle Li, Xiaolong Li, Xuan Li, Zhaoshuo Li, Zhiqi Li, Hao Liang, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Ming-Yu Liu, Sifei Liu, Zihan Liu, Hai Loc Lu, Xiangyu Lu, Alice Luo, Ruipu Luo, Wenjie Luo, Jiangran Lyu, Martin Ding Ma, Nic Ma, Qianli Ma, Dawid Majchrowski, Louis Marcoux, Miguel Martin, Qing Miao, Ashkan Mirzaei, Shreyas Misra, Kaichun Mo, Durra Mohsin, Hyejin Moon, Pawel Morkisz, Saeid Motiian, Kirill Motkov, Seungjun Nah, Yashraj Narang, Deepak Narayanan, Thabang Ngazimbi, Julian Ouyang, Shubham Pachori, David Page, Yatian Pang, Sehwi Park, Mahesh Patekar, Mostofa Patwary, Marco Pavone, Trung Pham, Wei Ping, Soha Pouya, Shrimai Prabhumoye, Varun Praveen, Delin Qu, Hesam Rabeti, Morteza Ramezanali, Marilyn Reeb, Xuanchi Ren, Kristen Rumley, Wojciech Rymer, Jun Saito, Yeongho Seol, John Shao, Piyush Shekdar, Tianwei Shen, Humphrey Shi, Min Shi, Stella Shi, Kevin Shih, Mohammad Shoeybi, Mateusz Sieniawski, Shuran Song, Alexander Sotelo, Amir Sotoodeh, Sunil Srinivasa, Vignesh Srinivasakumar, Bartosz Stefaniak, Rahul Heinrich Steiger, Shangkun Sun, Jiaxiang Tang, Shitao Tang, Yangyang Tang, Yue Tang, Tolou Tavakkoli, Kayley Ting, Krzysztof Tomala, Wei-Cheng Tseng, Jibin Varghese, Sergei Vasilev, Thomas Volk, Raju Wagwani, Roger Waleffe, Andrew Z. Wang, Boxiang Wang, Haoxiang Wang, Qiao Wang, Shihao Wang, Shijie Wang, Ting-Chun Wang, Yan Wang, Yu Wang, Rohit Watve, David Wehr, Fangyin Wei, Xinshuo Weng, Jay Zhangjie Wu, Kedi Wu, Hongchi Xia, Summer Xiao, Tianjun Xiao, Kevin Xie, Daguang Xu, Jiashu Xu, Mengyao Xu, Ruqing Xu, Xingqian Xu, Yao Xu, Dinghao Yang, Dong Yang, Hans Yang, Xiaodong Yang, Xuning Yang, Yichu Yang, Yurong You, Zhiding Yu, Hao Yuan, Simon Yuen, Xiaohui Zeng, Pengcuo Zeren, Cindy Zha, Haotian Zhang, Jenny Zhang, Jing Zhang, Liangkai Zhang, Paris Zhang, Shun Zhang, Xuanmeng Zhang, Zhizheng Zhang, Ann Zhao, Yilin Zhao, Yuliya Zhautouskaya, Charles Zhou, Fengzhe Zhou, Shilin Zhu, Yuke Zhu, Dima Zhylko, and Artur Zolkowski. Cosmos 3: Omnimodal world models for physical ai, 2026a. URL https://arxiv.org/abs/2606.02800.

NVIDIA, :, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman

Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezanali, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao Tang, Sameena Tasmeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. World simulation with video foundation models for physical ai, 2026b. URL https://arxiv.org/abs/2511.00062.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2nd edition, 2009. ISBN 978-0-521-89560-6. doi: 10.1017/CBO9780511803161.

Abdul Mohaimen Al Radi, Kunyang Li, Yuzhang Shang, Mubarak Shah, and Yu Tian. Aeroworld: Action-conditioned aerial video generation from inertial controls, 2026. URL https://arxiv.org/abs/2605.19728.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollar, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/45c1f6a8cbf2da59ebf2c802b4f742cd-Paper-Conference.pdf.

Jonathan Richens and Tom Everitt. Robust agents learn causal world models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pOoKI3ouv1.

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, 2025. URL https://arxiv.org/abs/2503.20523.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=rvUq3cxpDF.

Yiqing Shen, Aiza Maksutova, Chenjia Li, and Mathias Unberath. Counterfactual world models via digital twin-conditioned video diffusion, 2025. URL https://arxiv.org/abs/2511.17481.

Aman Sunesh, Allan Ma, and Siddarth Nilol. Modeling information blackouts in missing not-at-random time series data, 2026. URL https://arxiv.org/abs/2601.01480.

Yuanyuan Wang, Wenjie Wang, Kun Zhang, and Mingming Gong. Physics from video: Identifiability of time-invariant second-order ODEs under minimal trajectory conditions. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=xrXxqLadvS.

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4389–4398, 2024. doi: 10.1109/CVPR52733.2024.00420.

Sherry Yang, Yilun Du, Seyed Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 45210–45234, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/c4d66eae503694424123b93ac0fbaf17-Paper-Conference.pdf.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=VYOe2eBQeh.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y. Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T. Freeman. Physdreamer: Physics-based interaction with 3d objects via video generation. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 388–406, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72627-9.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=D5RNACOZEI.

Yunsong Zhou, Michael Simon, Zhenghao Peng, Sicheng Mo, Hongzi Zhu, Minyi Guo, and Bolei Zhou. Simgen: Simulator-conditioned driving scene generation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 48838–48874. Curran Associates, Inc., 2024. doi: 10.52202/079017-1548. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/57c5a7c83b056d74bc97b7db36bd3649-Paper-Conference.pdf.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9834–9844, 2025. doi: 10.1109/ICCV51701.2025.00917.

APPENDIX CONTENTS

A Notation, terms and proofs **18**

A.1 Proofs of Propositions 1–4 18

B The physics twin **19**

B.1 The display model 20

B.2 Dynamics 20

B.3 Reconstruction and command recovery 21

B.4 Acceptance tests of the twin 23

B.5 The search-based twin, and why it was replaced 24

C The dataset **26**

C.1 Pipeline stages 26

C.2 Footage, reading and labels 26

C.3 Actions: channels, types, units and sources 28

C.4 Sampling, coverage and the mixture 29

D Counterfactual pairs and rendering **30**

D.1 The v5.0 factory and rigs 31

E World models **32**

E.1 Cosmos 3 Edge: recipe 32

E.2 Cosmos-Predict2.5-2B: recipe and the diagnosed first attempt 33

E.3 Training curves 34

F CommandBench **34**

F.1 Sets and generation 34

F.2 Judge validation 35

F.3 Statistic 35

F.4 The loss probe 36

F.5 Real footage 36

F.6 Earlier instruments 36

G Results in full **37**

G.1 Command following, all sets, seeds and backbones 37

G.2 Ablations 37

G.3 Fidelity, second bench 39

G.4 The video-to-video route 39

G.5 Predict2.5 diagnostic 40

H Compute **41**

A NOTATION, TERMS AND PROOFS

Table 4: Symbols. Physical scalars are italic; world-model tensors are bold.

symbol	meaning
$o_{1:T}$	video frames; $o_{1:P}$ are the conditioning frames
$\boldsymbol{q}(t)$, $\boldsymbol{u}(t)$	vehicle state; commands (engines lit, throttle, thrust direction, surface deflections)
y_j, y_j^{obs}	a telemetry quantity and its displayed value, Eq. (1)
$\kappa_j,\ \delta_j,\ \Delta_j$	display low-pass time constant, delay and quantisation step of channel j
$\boldsymbol{c}$	observable context: conditioning frames, mission clock, flight phase, camera view
ι_k	command index of conditioning channel k (Definition 1)
$C(W)$, λ	coverage of a window of length W; mean command rate
$V,\ h,\ \gamma,\ r$	speed, altitude, flight-path angle, distance from the Earth's centre
$\boldsymbol{f} = (f_t, f_n)$	specific force along and normal to the velocity, Eq. (2)
$s = T/m$	thrust per unit mass
$\alpha,\ \sigma$	incidence (velocity to thrust axis); bank angle
$N,\ \eta$	number of lit engines; throttle
$T_{\mathrm{vac}},\ c_{\mathrm{vac}},\ A_e,\ p$	vacuum thrust, vacuum exhaust velocity, nozzle exit area; ambient pressure
c_{eff}	effective exhaust velocity $T/\lvert\dot{m}\rvert$
$D,\ N_{\mathrm{b}},\ \bar{q}$	axial drag, body normal force, dynamic pressure
k	crossflow–density scale of the entry aerodynamics
$\boldsymbol{x},\ \boldsymbol{\epsilon},\ \tau$	video latents, Gaussian noise, flow-matching time ($\tau = 1$ is noise)
$\boldsymbol{a}_t,\ \boldsymbol{\mu}_t$	conditioning vector fed to the world model and its validity mask
$f_\theta,\ G_\theta$	velocity network; sampler (conditioning frames, commands, seed ξ) $\mapsto$ video
R^a, R^b	renders of the two members of a counterfactual pair
follows	wins over all pairs on a CommandBench set (Section 6.1)

Terms.

- The *physics twin* is a JSBSim model of one flight whose state follows the broadcast telemetry.
- The *display model* is Eq. (1).
- *Command recovery* maps the reconstructed path to commands (Section 3.3).
- A *counterfactual pair* is two rendered flights of the twin from one initial state that differ in one command.
- Under the *robot form*, state cells are valid on row 0 only and commands on every row.
- Under *full-window state*, every valid cell is visible on every row.
- *Telemetry conditioning*, the baseline, builds the vector from the displayed telemetry without the twin (Section 5).
- A *variant* is one trained model compared in the tables, named by its training data and conditioning form (for example M0, H, H-full and H-real).
- TRUE, SWAP, SELF, NULL and STATE are the *conditions* of a pair.
- *CommandBench*, *the warm-pixel judge* and the *loss probe* are defined in Section 6.1 and Appendix F.
- We use *action* for whatever vector a world model is conditioned on, *command* for a quantity the vehicle's controller sets, and *state* for a consequence of the commands. The paper is about the distinction between the last two.

A.1 PROOFS OF PROPOSITIONS 1–4

Proposition 1. If $\iota_k = 0$ for all k then $\boldsymbol{a} = \psi(\boldsymbol{c})$ almost surely, so the joint law of $(\boldsymbol{c}, \boldsymbol{a})$ is supported on the graph $\Gamma = \{(\boldsymbol{c}, \psi(\boldsymbol{c}))\}$. The risk is an expectation under that law and depends on g only through its restriction to Γ; two functions equal on Γ have equal risk. $\tilde{g}$ equals $g^\star$ on Γ and is constant in $\boldsymbol{a}$. □

Remark 1 (Finite command index). For $\iota_k > 0$ the conditional law of a_k given $\boldsymbol{c}$ has spread $\iota_k \mathrm{Var}(a_k)$. The response of $g^\star$ to a_k is then constrained only within that spread, and the variance of any estimate of $\partial g/\partial a_k$ scales as $1/(\iota_k \mathrm{Var}(a_k))$. This is the positivity (overlap) condition of causal inference (Pearl, 2009), stated for conditioning channels. On a rocket flight, altitude and speed have $\iota \approx 0.01$. The mission-clock channel has $\iota = 0$ by construction and served as the estimator's positive control.

Remark 2 (Why the channel still lowers the training loss). On Γ, $\boldsymbol{a}_{1:T}$ is a function of the future state and hence informative about the future frames: $I(\boldsymbol{a}_{1:T}; o_{P+1:T} \mid o_{1:P}) > 0$ whenever the conditioning frames do not determine the future. A model can therefore lower its loss by reading $\boldsymbol{a}$, and metrics computed under the recorded $\boldsymbol{a}$ improve, without any constraint on its response off Γ. A linear read-out that recovers a_k well from video features shows the converse. Such a channel cannot inject information the video lacks; it can only select among videos.

Remark 3 (States are downstream of commands). In the causal graph $\boldsymbol{u} \to \boldsymbol{q} \to o$, a channel $\boldsymbol{a} = \phi(\boldsymbol{q})$ is a descendant of $\boldsymbol{u}$. Even with overlap, $p(o \mid \boldsymbol{a})$ is a conditional on an outcome of the command and differs in general from $p(o \mid \mathrm{do}(\boldsymbol{u}))$.

Proposition 2. For a Cox process directed by a random measure with Λ_W its mass on a window of length W, the number of points in the window is Poisson given Λ_W, so $\Pr(\text{no point}) = \mathbb{E}[e^{-\Lambda_W}]$. By stationarity $\mathbb{E}\Lambda_W = \lambda W$, and by Jensen's inequality $\mathbb{E}[e^{-\Lambda_W}] \geq e^{-\lambda W}$, with equality if and only if Λ_W is almost surely constant, that is, the process is Poisson on windows of that length. □

The ratio $(1 - e^{-\lambda W})/C(W)$ grows with the dispersion of Λ_W. For commands that cluster in bursts of duration b, windows with $W \gg b$ gain coverage at the rate of *bursts*, not of commands. This is why the measured shortfall widens with W.

Proposition 3. Let $(\boldsymbol{p}, m, T, \hat{\boldsymbol{b}})$ satisfy $m\ddot{\boldsymbol{p}} = T\hat{\boldsymbol{b}} + m\boldsymbol{g}$ and $\dot{m} = -T/c_{\text{eff}}$. Then $(\lambda m)\ddot{\boldsymbol{p}} = (\lambda T)\hat{\boldsymbol{b}} + (\lambda m)\boldsymbol{g}$ and $\frac{\mathrm{d}}{\mathrm{d}t}(\lambda m) = -(\lambda T)/c_{\text{eff}}$, so $(\boldsymbol{p}, \lambda m, \lambda T, \hat{\boldsymbol{b}})$ is a solution with the same path. With an aerodynamic force $\boldsymbol{F}$ that does not scale with m the first equation becomes $\ddot{\boldsymbol{p}} = s\hat{\boldsymbol{b}} + \boldsymbol{g} + \boldsymbol{F}/m$, and the scaled solution requires $\boldsymbol{F}/(\lambda m) = \boldsymbol{F}/m$, which fails by $(1 - \lambda^{-1})\boldsymbol{F}/m$. The symmetry is therefore broken in proportion to $|\boldsymbol{F}|/m$ relative to s, that is, to D/T. □

Under the throttle law $T = N(\eta T_{\text{vac}} - pA_e)$ the scaled solution has throttle $\eta' = \lambda\eta - (\lambda - 1)pA_e/T_{\text{vac}}$. The constraint $\eta' \leq 1$ bounds λ, and hence m_0, from above; an engine's minimum throttle bounds it from below. If a search's loss depends on the path alone, its minimiser is a curve in parameter space, and box bounds cut that curve at its ends. An acceptance rule of the form "no parameter on a bound" then tests a symptom, so we report identifiability directly instead (Appendix B.5).

Proposition 4. Take $\pi_{\text{render}} \geq \pi_{\text{real}}$ and write π_1, π_0 for them and $d \in \{1, 0\}$. Within a domain the inputs carry no information about e, and d is decodable, so the Bayes predictor from the inputs is a function of d. The one with the largest balanced accuracy is $\hat{e} = d$ (or its complement). Then

$$\Pr(\hat{e}{=}1 \mid e{=}1) = \frac{\rho\pi_1}{\rho\pi_1 + (1-\rho)\pi_0}, \qquad \Pr(\hat{e}{=}0 \mid e{=}0) = \frac{(1-\rho)(1-\pi_0)}{(1-\rho)(1-\pi_0) + \rho(1-\pi_1)}.$$

If $\pi_1 = \pi_0$ these are ρ and $1 - \rho$ and sum to one. Their sum is increasing in π_1 and decreasing in π_0, so it exceeds one exactly when $\pi_1 > \pi_0$. □

The share ρ changes the size of the shortcut but not its existence. Equalising π_d removes it. So does removing the features that announce the domain; in our data these were an out-of-vocabulary phase label on every render, render-only camera labels and a caption prefix. The probe shows that the shortcut is *available*, not that a world model uses it.

B THE PHYSICS TWIN

This appendix is the audit trail of Section 3. Every record it draws on is addressed by identity (configuration hash), never by recency.

B.1 THE DISPLAY MODEL

Model. Each displayed channel is $y^{\mathrm{obs}}(t) = Q_\Delta[(g_\kappa * y)(t-\delta)]$. The physical quantity y passes through $n \in \{1, 2\}$ identical first-order stages of time constant κ ($n = 2$ is critically damped), is shown δ later, and is quantised to the display step Δ. The informative samples are the display's *update instants*, at which the smoothed quantity crosses a display level. For a one-step change that level is the midpoint of the old and new values if the display rounds, and the new value if it truncates. A run of identical readings counts as one reading. For fixed (δ, κ) the model is linear in the coefficients of a small basis of acceleration histories (one per engine-count state, a constant, and gravity along the velocity). We therefore search (δ, κ) on a grid and solve for the coefficients linearly at each grid point.

Measured steps. On the four most recent flights (F10-F13) the altitude display steps by 0.1 km and the speed display by 1 km/h. Both are the smallest non-zero change of the raw digits. Even at the true step, a differentiation window of ± 3 s fails outright (incidence errors of 12–15°), which sets the scale of smoothing that any reconstruction needs.

Delays from first principles. With the delays left free, the kinematic residuals pass at any lag. The marginal likelihood is flat and irregular to ± 0.5 s (estimate $2.45/4.25$ s for a planted $2.10/3.95$ s). Two physical facts ground the delays.

1. *Rest at release.* The vehicle is at rest until release, so the displayed speed ramp must extrapolate to zero at release time plus the delay. A quadratic through the first seconds gives 1.54 s on the principal flight. Thrust builds over about two seconds after release, so the pin holds to about ± 0.3 s. A delay of 2.1 s would require 5.6 m/s one second after release, which a release acceleration of 3.6 m/s^2 cannot produce.
2. *Relative lag from the attitude indicator.* We scan the lag of altitude relative to speed. The indicator's nose elevation agrees best with the reconstructed flight-path angle at a relative lag of 1.85–2.0 s (rms 1.3–1.5°, against 2.8–5.2° at 0–1 s). We use 1.9 s, which gives 3.45 s for altitude.

An event-blind instrument, which does not know these values, supports them (Appendix B.4). On the pad the display shows one altitude step, a datum of about 100 m, which we subtract.

Other channels. The engine-state indicator is read from the frame, with a bracket of ± 0.02 s. The propellant gauge lags far more. We calibrated its lag on events free of venting and validated it on 1,980 synthetic fits. The booster gauge lags 13.5 s at lift-off on two flights, and the mass profile of Appendix B.3 independently prefers about 14 s. In the released labels the attitude-indicator, propellant and load-factor channels keep the display's clock and their lags are not measured.

Feed stalls. Reading the digits on every frame showed that the speed feed can *stall*. The shown rate falls, then overshoots until the display regains its line, while the altitude runs on. Read as dynamics, one such stall is a throttle of 0.66 and an incidence of $-48°$. The detector flags an interval in which the shown rate leaves its running level one way, then the other, within 3 s, and the two excursions cancel to within half of the first. A throttle change or a shutdown is one-sided and is kept. *Known answer.* With no stall planted, none is found, the known throttle reduction at maximum dynamic pressure, the throttle-down and the main cut-off are not flagged. Three planted stalls are each found at the planted time plus the display delay. On the principal flight the detector finds exactly one stall. The catalogue gaps in two flights both sit on a stall.

Clock. The mission clock drawn on the display leads the frame's own time by a phase that is stable over a flight. On every flight of the current graphics it re-synchronises once, about ten seconds after lift-off, and readings stamped before then are about 0.5 s early. Until we found this, one flight's first 30 s could not be reconstructed.

B.2 DYNAMICS

Translational model for reconstruction. In the Earth-fixed rotating frame,

$$\ddot{\boldsymbol{p}} \;=\; \boldsymbol{g}(\boldsymbol{p}) + \boldsymbol{c}_{\mathrm{f}}(\boldsymbol{p}) + \boldsymbol{c}_{\mathrm{c}}(\dot{\boldsymbol{p}}) + \boldsymbol{f}, \qquad m\,\boldsymbol{f} \;=\; T\,\hat{\boldsymbol{b}} + \boldsymbol{F}_{\mathrm{aero}}(h, V, \hat{\boldsymbol{b}}), \tag{5}$$

with $\boldsymbol{g}$ the gravitational acceleration including the J_2 term, $\boldsymbol{c}_\mathrm{f} = -\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \boldsymbol{p})$ and $\boldsymbol{c}_\mathrm{c} = -2\,\boldsymbol{\Omega} \times \dot{\boldsymbol{p}}$. The cross-track component is unobserved. The launch azimuth enters only through the frame terms; moving it across its bracket (85–110°) changes the recovered incidence by $0.02°$. The atmosphere is the 1976 US Standard Atmosphere.

Propulsion. For N identical engines at throttle η, $T = N(\eta\, T_\mathrm{vac} - pA_e)$; the pressure loss is not throttled. Mass flow is $\dot{m} = -N\,\eta\, T_\mathrm{vac}/c_\mathrm{vac}$. Substituting $T = s\,m$ gives Eq. (4). A stage with several engine sets i on one throttle (the upper stage's sea-level and vacuum engines) has $T = \eta \sum_i N_i T_i - p \sum_i N_i A_i$ and $\dot{m} = -\eta \sum_i N_i T_i / c_i$.

Aerodynamics in powered flight. Axial drag is $D = \bar{q}\, C_A(M)\, S$, with C_A tabulated in Mach number M. The body normal force is $N_\mathrm{b} = \bar{q}\, S_\mathrm{p}\, C_c(M) \sin\alpha |\sin\alpha|$ on the planform area S_p. At $5°$ of incidence through maximum dynamic pressure, N_b is about 7 % of the normal component of thrust. Equation (3) is implicit in m and, through N_b, in α. Six fixed-point iterations of (3)–(4) converge.

Effective exhaust velocity from the path. At constant throttle above the sensible atmosphere T is constant and $s = T/m$, so $\dot{s} = -T\dot{m}/m^2 = s^2/c_\mathrm{eff}$, that is

$$\frac{\mathrm{d}\ln s}{\mathrm{d}t} = \frac{s}{c_\mathrm{eff}}. \tag{6}$$

The growth of thrust per unit mass at constant throttle thus measures c_eff without knowing thrust or mass. We found no published statement of this estimator.

Forward model. The forward re-flight integrates the full rigid-body equations in JSBSim (Berndt, 2004) at 120 Hz, with control at 10–20 Hz, a rotating Earth and J_2 gravity. The model carries the attitude quaternion, body rates, per-engine thrust and gimbal, propellant tanks with centre-of-mass travel, and the aerodynamic surfaces. At staging, twelve state properties pass between two vehicle instances. No aerodynamic model of the vehicle is public. We author drag and crossflow tables from published geometry, refine them per hardware generation, and share them across the flights of a generation. A new flight then needs only configuration: lift-off anchor, display clock phase and delays, the frame-read engine schedule, launch site, azimuth bracket and payload mass.

B.3 Reconstruction and command recovery

Smoother. The smoother is an extended Kalman filter with a Rauch–Tung–Striebel (RTS) backward pass (Sunesh et al., 2026). Its state is $\boldsymbol{x}^\mathrm{r} = [d, h, \dot{d}, \dot{h}, f_d, f_h]$ in the local-vertical frame of the ground track. The specific force is a random walk with spectral density q, *opened* to a large value in a short interval around each change of the engine-state indicator. On ascents q is 0.1 $\mathrm{m^2 s^{-5}}$. A stretch that refuses more than 2 % of its readings is re-run once at $q = 1$, and the better run is kept. The measurements are $|\dot{\boldsymbol{p}}|$ and h, at time stamps shifted by each channel's delay. Three details were each forced by an observed failure. (i) *Relinearisation.* A single forward pass diverges for a loose walk, with thrust per unit mass in error by more than 200 %. Relinearising the speed measurement about the smoothed path over three passes holds for $q \in [0.01, 1]$. (ii) *Mirror branches.* Because speed is $|\dot{\boldsymbol{p}}|$, the filter can settle from rest on the path reflected up-range, or below the pad. Both branches leave s and mass correct and flip the sign of α. They are an exact symmetry up to the Coriolis term. During the launch the branch is imposed on the linearisation, not on the state (repair (iv) below). (iii) *Gating.* Gross misreads are gated, with re-acquisition. Posterior paths drawn by backward sampling are pushed through the closure, so every recovered quantity has a 68 % band. On the known answer these bands are conservative (standardised error below one).

Whole flight in one pass. Every stretch in which the display shows both channels is smoothed as one run per vehicle, in about a minute of computation per flight. Coast is free fall; the walk is closed above 100 km when no engine is shown. On the four most recent flights one run covers the upper stage's coast, entry, descent and landing burn. It ends within one display step of the surface, at 4–17 m/s. The search-based twin needed four chained fits for the same span and lit its landing burn 14.5 s late. A vehicle's rows end where its display panel passes to the other vehicle. During the upper stage's ascent burn the display draws *no altitude* for that stage, for almost five minutes. The

fixed-interval smoother treats the gap as the two-point boundary problem it is, with anchors on either side. Its altitude spread grows to ± 22 km in the middle of the gap, while speed keeps the along-track acceleration pinned. Labels are masked per channel on their own spread.

Second algorithm and the choice of differentiator. The algebraic inversion needs $\dot{V}$ and $\dot{\gamma}$ from noisy, quantised channels. On the known answer we compared a local polynomial, a cubic Savitzky–Golay filter and a total-variation-regularised derivative, at half windows of 5, 8 and 12 s. Savitzky–Golay at ± 12 s was best (thrust per unit mass 0.20 %, mass 0.013 %, incidence 0.61°). The total-variation scheme was worst (0.65 %, 1.80 %, 4.62°), a verdict on our implementation of it. On the principal real ascent the two algorithms differ by 0.63 % in s and 1.08° in α, within their combined spread. A windowed differentiator smears a thrust step. The smoother, whose noise opens only at engine events, is therefore primary.

Known-answer test. We pass a simulated ascent through a planted display (delays 2.10 and 3.95 s, the measured steps) at the broadcast's own reading times. A synthetic attitude indicator runs at 1 Hz with 0.6° noise. The reconstruction is blind to the throttle and mass history (Table 5). The attitude indicator is the compatibility check that refuses the third row. The kinematic residuals pass the first three rows, so they alone would not have caught a missing display model.

Table 5: Known-answer test of the reconstruction: error against the planted truth.

variant	delays (s)	s	mass	throttle (abs.)	α
delays estimated	2.45 / 4.25	0.25 %	0.040 %	0.0021	0.43°
delays as planted	2.10 / 3.95	0.24 %	0.024 %	0.0021	0.50°
no display model	0 / 0	2.00 %	0.605 %	0.0111	3.24°
digits read every frame	2.10 / 3.95	0.09 %	0.024 %	–	0.21°

Mass identification. With $s(t)$ fixed by the path, Eq. (4) gives a mass history $m(t; m_0)$ for any lift-off mass m_0. The ascent alone cannot fix m_0 (Proposition 3); a planted -2 % throttle fault buys back -2.0 % of mass on 12 of 12 flights. A second equation that bypasses thrust breaks the degeneracy. The booster's propellant gauge reads a fraction of a fixed capacity, $f(t+\delta) = g((m(t) - R)/C)$ with $g(f) = f + \kappa f(1-f)$. Here R is everything the gauge does not show, C its full scale, δ its display delay and κ a volume-height bow. The slope of f against the closure's mass flow fixes m_0/C, and its level fixes R/C. A joint maximum-a-posteriori fit across flights shares dry mass and capacity per booster generation. It takes load fraction, payload and ring per flight, and adds the landing chains as further likelihood rows. We use the fit only while leave-one-out passes against the four flights with an independent mass ledger. The median error is 0.54 %, with 4 of 4 inside 2σ. The uncalibrated variant reads 1.30 % and is reported as the bound. The identified masses are 4,843–4,970 t on the first generation, 5,098–5,331 t on the second, and 5,565 and 5,497 t on the third (Table 1). The last lies inside the published 5,540–5,670 t. These numbers are close to reality as newer vehicle versions have increased fuel capacity and carry more payload.

A positive control across instruments. We reconstruct a second flight by configuration alone. Its throttle reduction at maximum dynamic pressure comes out at 0.732. The slope of that flight's propellant gauge over the same seconds gives 0.738where the gauge shares nothing with the kinematics.

Entry. Hypersonic entry (Mach above 2.5, below 90 km) is split into 10 s bins. In each bin, drag and vertical lift per unit mass and the attitude indicator's nose angle are solved for (α, σ, k) by bounded least squares, with $\alpha \in [20^\circ, 85^\circ]$, $\sigma \in [0^\circ, 89^\circ]$ and $k \in [0.3, 3]$. Widths come from the Jacobian at the measurement noise. A planted $(52^\circ, 35^\circ, 0.8)$ is recovered from three starting points. On the principal flight the median incidence is 54.2°, with a width of 2.8°. The attitude indicator is reproduced at 0.27° rms, the noise floor of its reader. Held out, the re-flown entry matches speed to 1.87 % and altitude to 1.08 %. For comparison, a nine-parameter search had failed to optimise this part on every seed. Flap deflections follow from the moment balance at the recovered attitude. Their spread is the trim at either end of the incidence width. No flap has yet been compared with a view of a flap.

Table 6: Where the inversion degenerates, and the closure used.

regime	what fails	closure
near-vertical flight ($\cos\gamma \to 0$)	normal direction undefined; direction of the path unobservable at the display's altitude step	local-vertical chart; along-track equation only; incidence reported as missing, not as a wide band
flight-path angle from two channels with different delays	$\sin\gamma = \dot{h}/V$ mixes clocks (scatter of 9° at one landing)	remove each channel's display model before combining
coarse altitude display	$\ddot{h}$ dominated by quantisation	smoother; long differentiation window
coast	$\boldsymbol{f} = 0$: nothing to invert	free fall between pinned end states
altitude not displayed	direction unobservable from speed alone	two-point boundary problem; spread reported; labels masked on it
cross-track motion	unobserved	azimuth bracket; effect on incidence 0.02°
unpowered, high incidence	k = crossflow $\times$ density not separable; sign of bank unobservable	product reported; positive branch flown and flagged

B.4 Acceptance tests of the twin

Every check below was fixed before it was applied. Each instrument was first run on a case with a planted answer.

Table 7: Checks applied to a reconstruction. "Known answer" is what the instrument was shown to see before it judged a real flight.

check	bar	known answer
data compatibility	channels mutually consistent after the display model; attitude indicator against reconstructed path	planted delay recovered; absence of the display model refused
reconstruction	$s \le 1\,\%$, mass $\le 1\,\%$, $\alpha \le 0.5^\circ$ on a simulated ascent at display resolution	Table 5
event clock	event-blind kinematic time of a thrust step within the display-delay band of its frame-read time	delay error followed one for one
two algorithms	smoother and algebraic inversion agree within their combined spread; disagreement reported, not averaged	same simulated ascent
forward re-flight	two conditions of Section 3.4	two planted faults refused
label clock	every engine event in the labels within 0.2 s of its frame-read time	measured 0.017 s

The two conditions and its planted faults. Table 8 gives the principal flight and a second flight. The median residuals alone accept the principal flight's zero-incidence fault. Its median altitude residual is 1.04 %, while the burn ends more than 1.5 km low and 3° shallow. The end-state bound alone accepts the second flight's fault, where reconstruction and fault fly nearly the same path. Together the two parts refuse all planted faults.

Table 8: Open-loop re-flight to the end of the ascent burn: state error at burn end against the reconstruction, and the verdict of the two conditions.

flight	variant	end altitude	end speed	end γ	verdict
principal	reconstruction	-243 m	$+3.5$ m/s	-0.17°	accepted
	thrust along velocity	$-1{,}547$ m	–	-3.1°	refused
	throttle $-2\,\%$	–	-36.4 m/s	–	refused
second	reconstruction	$+618$ m (1.08 %)	-1.8 m/s	$+0.9^\circ$	refused, by 0.08 %
	thrust along velocity	$-1{,}331$ m	–	-2.2°	refused
	throttle $-2\,\%$	–	-40.5 m/s	–	refused

A physical finding from the re-flight. After maximum dynamic pressure on the principal ascent, the reconstruction has an incidence of $+3$ to $+4^\circ$. The attitude indicator, as we model it, puts the nose within 2° of the velocity. Varying azimuth, altitude scale, speed scale or walk density one at a time does not remove the difference. The algebraic inversion, which has no dynamics prior, finds the same value. Flown open loop from one state, the six-degree-of-freedom model follows the display to the end of the burn with the reconstructed incidence. With thrust along the velocity it sinks more than 1.5 km. Our attitude-indicator model was validated under a zero-incidence assumption. We therefore use the indicator only for the *relative* display delay and regard its absolute elevation as unresolved.

A withdrawn instrument. Our first event-clock check compared the frame-read time of a thrust step with the time of the step in the *smoother's* specific force. It reported agreement to 0.1 s. The smoother's noise opens at the frame-read time, however, so its step sits there by construction, and the agreement measured nothing. The replacement takes the path's clock from an event-blind differentiator (Savitzky–Golay, ± 2 s, no engine information). On a known answer, assumed delays wrong by -1, 0 and $+1$ s move the kinematic minus frame-read time to $+1.69$, $+0.69$ and -0.31 s. The clock thus follows a delay error one for one and the offset at zero error is the model's own thrust decay. On real flights the instrument places the main cut-off -0.29 s and 0.00 s from its frame-read time. This supports the delays of Appendix B.1 with an instrument that does not know them. Single-engine shutdowns are below its noise.

Four clocks on one event. Four clocks time the landing-burn ignition of the principal flight. The engine-state indicator shows first light at T+3,902.35 s and three engines lit at 3,903.35 s. The kinematic onset of thrust is at 3,904.0 s (3,903.25–3,905.25). The plume is visible to the camera at 3,905.1–3,907.7 s. The chained search-based twin ignites at 3,916.83 s. The three physical instruments order as physics requires (light-up, measurable thrust, visible plume), within three seconds. The search-based twin lies 12.8 s outside all of them, because altitude triggers its ignition and passes on its error. In factual replays we therefore fly every burn on the measured schedule, and the command label is the engine-state indicator. The visible plume is kept as a property of the *view* now being 1.7–5.4 s late, by a different amount per camera.

Status by flight. All 12 flights are reconstructed (49/49 stretches). The stack ascent passes the two conditions on every flight, and the throttle fault is refused on every flight. The zero-incidence fault is refused on 10 flights and is uninformative on IFT-5 and IFT-6 (planted-fault separation below 1). Eight landing burns pass with a recovered throttle, five on the booster and three on the upper stage. Two do not: the F13 upper-stage burn (rms 10.9 m/s) and the IFT-3 booster burn, where a 5 s burn cannot separate a $-2\,\%$ throttle. Three upper-stage burns have no engine indicator in view. On two flights the display omits the upper stage's altitude for the whole ascent burn. That altitude is then unobserved, and its throttle rests on the gauge alone. Entry attitude is reconstructed by trim on the four most recent flights and is not validated. Figure 3 shows the per-flight verdicts, the identified masses and one flight's ascent.

Instruments withdrawn or repaired. (i) The first event clock measured nothing, as described above. Its event-blind replacement reads $+1.69/+0.69/-0.31$ s for planted delay errors of $-1/0/+1$ s. (ii) The first acceptance bar divided the residual by the displayed value. On early flights the 1 km altitude step alone then came to 20–30 %, and four reconstructions were refused for the display's resolution. Scoring in units of the reading's own spread accepted them with no change to the physics. (iii) Era-B display delays had been placed on a half-sample lattice. The crossing instrument now measures them per flight, with a standard deviation. (iv) The smoother's launch branch once reflected the pad 2.9 km below ground. The reflection now acts on the linearisation only. (v) One label build shipped with a whole layer missing, and every gate improved. Identities now cover label content. A known-answer run or a planted fault found each defect, and the repository's experiment log records each one.

B.5 The search-based twin, and why it was replaced

Our first twin was an output-error fit. A forward simulation carried per-phase parameters: propellant load, added dry mass, a drag scale and throttle knots on ascent; per-engine-set throttles on the booster's return; and a crossflow scale on descent. The loss was the mean squared relative error of

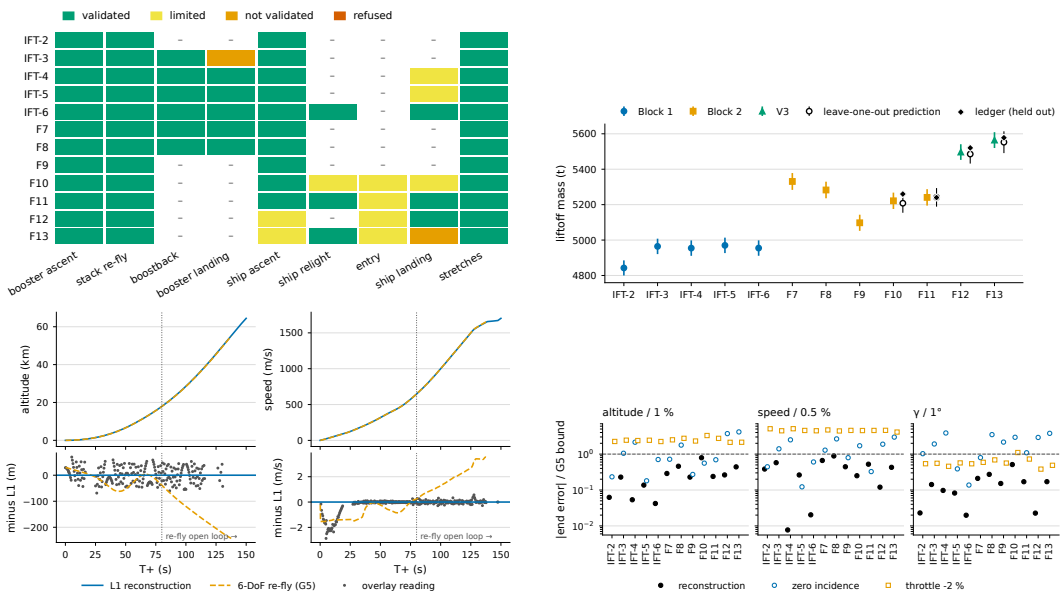


Figure 3: Twin acceptance on 12 flights. Top left: verdict per flight and phase. Top right: identified lift-off mass per flight with its σ, by vehicle generation. Bottom left: one flight's ascent, re-flight against the displayed telemetry through the display model, with the two planted faults. Bottom right: end-of-open-loop altitude, speed and flight-path angle against their bounds.

speed and altitude *seen through the display model*, in 2 s bins. The covariance-matrix-adaptation evolution strategy (CMA-ES) (Hansen, 2023) searched the unit box, with three seeds per phase. Fit and held-out windows were fixed in code before fitting. Held out, the twin matched each of one flight's four phases within 2 % when the phases were taken separately. We replaced it for four measured reasons.

1. The valley of Proposition 3. On a second flight, three seeds ended at drag scales of 2.00, 1.40 and 1.98 and added dry masses of 60.3, 82.6 and 74.0 t. Their losses were identical to five digits, with three parameters on bounds. The same optimiser recovers a planted interior optimum on three of three seeds, with a largest error of 0.06 % of the range, so the optimiser is not at fault. A rule of "no parameter on a bound" had to tell two kinds of bound apart. At one, the model has stopped responding and every probed loss is identical, so the fit is not evidence. At the other, the model responds monotonically, which marks a legitimate physical limit. The rule is a symptom test for a search, and the inversion has no search to test.

2. The display absorbed into parameters. A first round scored raw digits at zero delay. The parameters absorbed the lag, and held-out altitude error was 8.9 % raw and 15.4 % through the display. With the display model inside the loss, the same flight reads 2.2 %.

3. Free clocks. Per phase, the fitted twin's delay against the display was within 2 s on ascent and coast and +15 s at the landing. Events flown on the measured schedule were exact to 0.1 s. Events triggered by state inherited the state's error. Chained phases were flown outside the windows on which they were fitted. The landing's held-out error of 0.5 % alone became 24 % in speed when flown in the chain.

4. Cost and transfer. A nine-parameter entry fit cost 4 h and failed on every seed, and each new flight meant new fits. The inversion reconstructs a whole flight by configuration in about a minute.

The search-based twin remains the source of the vehicle tables that the inversion and the forward model share: engine law, dry masses, and drag and crossflow tables. It is also the source of the landing phase from which this paper's counterfactual pairs branch (Appendix D).

C THE DATASET

The release is `rocketwm-v5.0`, with 5,061 real clips of 12 flights and 2,052 rendered clips. Every file is content-addressed under one identity hash. The hash covers the base footage, every label layer, the registry, the split and the build configuration. Table 9 and Figure 4 give the pipeline stage by stage.

C.1 PIPELINE STAGES

Table 9: Stages of the data pipeline, the method of each, and what was measured.

stage	method	measured
acquisition	official broadcasts of 12 flights from public platforms; licence tier recorded per clip	37 h kept across vehicles; one family used
normalise	constant 24/30 fps, x264, 16 kHz mono audio	–
shots, clips	PySceneDetect adaptive detector; 3–10 s clips, 8 s windows at 6 s stride	–
filters	blur, exposure, black frame, motion, letterbox; static-countdown decimation	–
critic	vision–language model on four frames per clip; only the critic may demote a crop	rule agreement with the critic 17 % and 3 % for two heuristics it replaced
dedup, appearance	pHash (summed Hamming ≤ 12 over 8 frames) and Cosmos-Embed1 cosine ≥ 0.92	duplicate ratio 0.013
dedup, identity	audio landmark offset, displayed-clock RANSAC, telemetry agreement	30/30 synthetic pairs within 0.25 s; 1/10 hard negatives matched
panes, clean crop	pane layout at 1 fps with hysteresis; 1280×704 crop centred on the vehicle (Grounding DINO and SAM 2.1, 5 frames per segment); display band dropped	146 of 5,227 candidates without a clean crop
display reading	digits by PaddleOCR at 2 fps; engine indicator by per-engine luminance at calibrated centres, re-read on every frame; gauges by radial sampling; attitude indicator by matched drawing or a learned ResNet-18 rotation head	exact on speed, altitude, clock (100 %); planted 3 s shift detected; gauge median error 3.03 $\rightarrow$ 0.14 pt
state estimation	Hampel filter (window 11, 3σ), constant-acceleration Kalman filter and RTS smoother, 10 Hz grid, gaps over 0.75 s masked	–
events	change-points of indicator, clock, subtitles and audio, refined to one frame; lift-off anchor per flight	bracket 0.03 s after refinement
physics twin	Section 3; per-flight 10 Hz table with σ	Table 1
label layers	displayed, delay-shifted, engine indicator, gauge, twin; resolved by trust with refutation	204/204 events within 0.2 s
registry, gates	roles per channel by measured coverage and visibility; 21 gate families	2 of 142 bars over, both bounded
rendering	Unreal Engine 5.8.2 replay of the twin tables and the counterfactual factory; four rigs	306 episodes, 720 pairs

C.2 FOOTAGE, READING AND LABELS

Footage. The 5,061 real clips last 3–10 s (11.1 h at 30 fps). They come from the official broadcasts of 12 flights of one aerospace vehicle family (SpaceX's Starship), across three generations of on-screen graphics. The programme's first flight is absent because its display shows no telemetry. Near-duplicate groups are resolved within a flight, never across flights. Each clip exists with and without the display region, and models train on the version without it. Camera view and the visibility of each actuator are recorded per clip from opened frames on the four most recent flights, and flagged as unjudged on the rest.

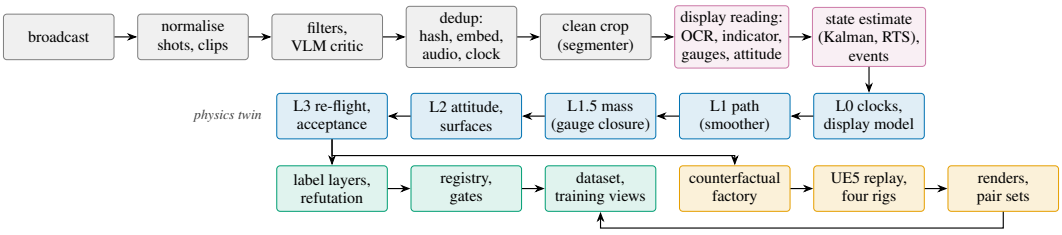


Figure 4: The data pipeline. Top: footage to display readings and state estimates. Middle: the physics twin, one direction of flow from measurement models to the accepted re-flight. Bottom: label layers and registry into the dataset; the twin also feeds the counterfactual factory and the renderer, whose clips join the dataset as a declared second domain.

Reading the display. Per broadcaster and graphics generation, a layout registry gives the fractional region, type and unit of each field. PaddleOCR reads the digits. The engine-state indicator is read from the luminance of each engine's circle at calibrated centres. A reading is accepted only where a blob lies within two radii of a calibrated centre; this closed a real fabrication on aerial shots before launch. Propellant arcs are read by maximum-luminance radial sampling, and the attitude indicator by the principal axis of its silhouette. After staging the display has two panels, one per vehicle, read as two tracks. Readings pass a Hampel filter and a constant-acceleration Kalman filter with an RTS smoother onto a 10 Hz grid. Gaps over 0.5 s are masked, not interpolated. Flight events are fused from telemetry derivatives, engine-indicator transitions, subtitle keywords and audio onsets (peaks of librosa's onset-strength envelope). Audio serves as corroboration only.

A known-bad test of the reader. On hand-read banks from three graphics generations, the reader is exact on speed, altitude and clock (100 %, no false positives). A scorer that can only return 100 % is worthless. We therefore re-scored the same banks with every time stamp shifted by 3.0 s. Exactness fell to 5 %, 40 % and 0 %, and the clock error landed on the planted shift. One generation's bank covers a single flight.

Labels are layers. The five layers are displayed readings, readings shifted by the flight's measured display delay, the engine-state indicator re-read on every frame, the propellant gauge, and the twin. A cell resolves to the highest-trust source whose confidence passes. Ties never average, and a layer may refute a lower cell. The release refutes 7.6 million cells. Of these, 5.5 million are booster readings held across a display freeze and 1.4 million a held booster engine indicator. A further 0.2 million lie on clips whose clock is more than 1 s off the broadcast's, and 2,880 are a fabricated engine reading. A single-producer rule takes every engine cell of a clip from one source (2.9 million cells on 2,469 clips). In million cells, the resolved sources are kinematic 6.0, displayed 2.8, engine indicator 2.0, observed 1.7 and trim 0.6. Every cell ships with its σ (5,058 clips). Events, 16 kHz audio, the raw and smoothed readings and the per-flight twin tables ship beside the clips.

Label clock. Against the twin's whole-flight tables, the displayed speed and altitude of an earlier release trailed the vehicle by 2.25 s and 4.25 s. On the pixel clock, every engine event in the labels falls within 0.2 s of its frame-read time (204/204, median 0.017 s). On the early graphics generation, label error against the display on the same frames fell from 1,228 to 117 m in altitude (median) and from 6.6 to 1.15 m/s in speed. On the corpus, the twin's σ scale against the readings is 0.76 for altitude and 1.12 for speed, against 3.61 and 3.63 one release earlier (Figure 5).

Split. One flight (IFT-4, 395 clips) is held out entirely. On the flight whose landing burn is intervened on, the held-out set is the union of fixed lists (the evaluation bench and the clips of the real-frame tests). It is closed under parent-clip and near-duplicate relations and holds 85 clips, leaving 564 trainable. A guard fails any export that contains a held-out clip, and no bench clip or IFT-4 clip is in any training variant. However, 93 % of clips overlap another clip of the same flight in mission time (34 % from another camera), and 53 of the 85 held-out clips overlap a training clip. A split by flight (IFT-4 and F13 held out, 21 % of clips) is therefore the stronger protocol.

Release. The release contains labels with σ and provenance, derived telemetry, per-flight twin tables and mass ledgers, the rendered clips and counterfactual pairs, the registry, the gates and the code. Broadcast video is not redistributed and clips are located by source identifier and time range.

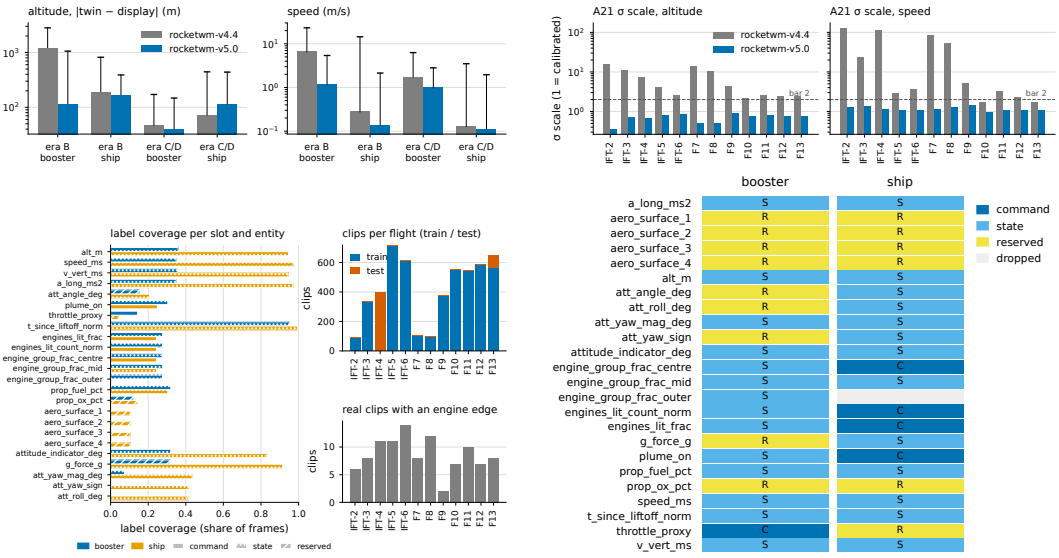


Figure 5: The dataset. Top left: label error against the display on the same frames, by graphics generation and vehicle, before and after the twin re-labelling. Top right: the twin's σ scale against the readings per flight. Bottom left: clips per flight and clips containing an engine change. Bottom right: the registry, channel by vehicle, coloured by role.

C.3 Actions: channels, types, units and sources

Table 10 lists the 24 channels per vehicle, one row per video frame in the release and 10 Hz in the twin tables. The registry types each channel per vehicle as *command*, *state* or *reserved*. A command is scored only where (a) it is labelled on at least 5 % of the target clips, (b) a camera view has been shown, by opening frames, to resolve its effect, and (c) its excess predictability from the flight phase is below 0.10. Only an aft-facing hull view resolves which of the booster's 33 engines are lit, and that view appears in a handful of clips. The booster's engine channels are therefore state. The upper stage's throttle (labelled on 4 % of its clips) and the flaps (by trim, unvalidated) are reserved. They ship, but are withheld from the input by default. The vector width is fixed, so a change in roles does not change a network.

Tensor forms. For Cosmos-Predict2.5-2B a window of 93 frames yields 92 rows of 2×50 values. For each of two vehicles a row holds the 24 values, their 24 validity bits, a vehicle-role code and a presence flag. Reserved cells stay in this tensor with validity 0. For Cosmos 3 Edge a row holds only the live cells, values then validity, with commands before state and the booster before the upper stage. The row is 62 wide under the training registry and 64 wide under v5.0, with one row per action step at 10 Hz. One function, shared by training, rollout and the loss probe, applies the conditioning form to the validity bits when a window is drawn, never to the values. Full-window state keeps every valid cell, the robot form keeps state valid on the window's first row only, and a third form masks state on every row.

Observability classes. Engine on/off by ring is *measured* from the frame (± 0.02 s). Thrust per unit mass, thrust direction in the vertical plane and, with the external mass scale, throttle are *algebraic* (Eqs. 3–4). Surface deflections and slow gimbal come *through trim*, from the moment balance at the recovered attitude. A trimmed polar can be wrong without the trajectory showing it. Three independent validations are planned and not done: a flap's hinge angle where one is in frame and the apparent motion of hull and horizon in a camera mounted *on* a flap, which is the flap's own motion,

Table 10: Action channels of `rocketwm-v5.0`. Each vehicle cell gives type and coverage. Type: C command, S state, R reserved, – absent. Coverage is the share of the vehicle's clips with at least one labelled frame. Source is the dominant label layer.

#	channel	unit	normalisation	booster	upper stage	source
0	altitude	m	symlog, 250 km	S, 0.36	S, 0.95	twin (kinematic)
1	speed	m/s	symlog, 8,000	S, 0.35	S, 0.97	twin
2	vertical speed	m/s	symlog	S, 0.36	S, 0.94	twin
3	longitudinal acceleration	m/s^2	/60	S, 0.35	S, 0.97	twin
4	body angle from vertical	deg	/180	R, 0.15	S, 0.21	twin (trim, kinematic)
5	any engine lit	0/1	–	S, 0.30	**C**, 0.25	engine indicator
6	throttle	fraction of rated thrust	–	**C**, 0.14	R, 0.04	twin (observed, kinematic)
7	time since lift-off	s	/5,400	S, 0.95	S, 1.00	clock
8	engines lit, fraction	–	–	S, 0.28	**C**, 0.24	engine indicator
9	engines lit, count	count / own engines	–	S, 0.28	**C**, 0.24	engine indicator
10	inner ring lit, fraction	–	–	S, 0.28	**C**, 0.24	engine indicator
11	middle ring lit, fraction	–	–	S, 0.28	S, 0.24	engine indicator
12	outer ring lit, fraction	–	–	S, 0.28	–	engine indicator
13	propellant, fuel	%	/100	S, 0.32	S, 0.30	gauge
14	propellant, oxidiser	%	/100	R, 0.12	R, 0.14	displayed
15–18	aerodynamic surfaces (fwd L/R, aft L/R)	deg	/90	R	R, 0.10	twin (trim)
19	attitude-indicator angle	deg	/180	S, 0.32	S, 0.83	displayed
20	load factor	g	/10	R, 0.32	S, 0.92	twin
21	yaw magnitude	deg	/90	S, 0.07	S, 0.43	displayed
22	yaw sign	± 1	–	R	S, 0.41	displayed
23	roll	deg	/180	R	S, 0.41	displayed

and the closed-loop re-flight. Reaction-control thrusters, fast gimbal motion, the sign of bank, and any split of k into crossflow coefficient and density are *not recoverable*.

One channel, two quantities. Earlier experimentation on label version computed the lit-count channel as a fraction of the vehicle's own engines on real clips, but of the fleet's largest engine count on renders. Three of six lit engines then read 0.50 on a real clip and 0.09 on a render. A mixed real and rendered training set fails silently on such a mismatch. The two definitions are now one.

C.4 Sampling, coverage and the mixture

Coverage. A command is a persistent change of the lit-engine count. It must hold for 1 s and must not straddle a reading gap of more than 2 s. Changes within 2 s merge, so a staged shutdown is one command. These constants were fixed before the numbers were read, and the detector found twelve of twelve independently anchored mission milestones. Table 11 compares coverage under uniform window placement with the constant-rate prediction at the same mean rate (one command per 166 s). Event-centred placement puts a command in every window by construction. The corpus admits 303 such windows, which bounds what real footage alone can teach.

Table 11: Share of uniformly placed windows that contain a command.

W (s)	measured	constant rate	shortfall
3	0.88 %	1.79 %	2.05×
12	2.85 %	6.97 %	2.45×
121	10.5 %	51.8 %	4.9×

The sampler. A window is a pure function of the resolved labels and a seeded generator, reproducible from (clip, seed, epoch). The classes *change*, *on*, *off* and *unlabelled* carry weights 0.35, 0.20, 0.20 and 0.25. Within a class, the rare flight phases (landing burn, entry, staging) are up-weighted by 3, 3 and 2.

Offset first. Let a clip of L frames contain a change at frame c, and let windows have n frames. The feasible offsets of the change inside a window form the set $\{\, o : 0 \le c - o \le L - n \,\}$. This interval is short and one-sided when c lies near an end of the clip. Drawing the start uniformly among windows

that contain the change draws o uniformly on *that clip's* interval. Pooled over clips, offsets near the window's ends are then over-represented. Drawing o uniformly on $[0, n)$ first, and passing to another class when the clip cannot serve it, makes every offset equally likely to be requested. The measured peak-bin share of the pooled offset histogram is 0.158 start-first and 0.130 offset-first.

One feasible start. If $L = n$ the feasible set is a single window, and no sampler can help. Our first renders were cut this way, with every ignition at one of three offsets. The peak-bin share was 0.333, and the window clock alone predicted engine state at a balanced accuracy of 0.962. Re-cut to 8 s, the same renders score 0.089 and 0.500. A regression of the window mean on context is blind to this defect, since $R^2 = 0$ when the clip is the window. An event channel is therefore judged on the offset histogram and the clock probe, not on R^2 alone.

Achievable, not configured. The sampler's realised class mix is scored against the mix the corpus can supply, not against the configured weights. Otherwise a sampler would fail for the corpus's poverty.

The context probe. The probe measures what a model could learn without reading the command. A gradient-boosted classifier (scikit-learn's `HistGradientBoostingClassifier`) predicts the engine state at a window-relative time from that time, the flight phase and the camera view, with the flight held out. Chance is 0.5. The probe was shown to see the fixed-offset defect above before it was used.

The mixture. Table 12 gives the observation behind Proposition 4 and its current state. Three features announced the domain: a phase label used on every render and absent from the real vocabulary, render-only camera labels, and a caption that began with the same three words on every render. The caption is a conditioning input. The phase label was a defect and is corrected. The other two are retained by decision, and the probe is reported with them.

Table 12: Balanced accuracy of predicting engine state from context alone (chance 0.5).

training set	probe
real clips only (uniform windows)	0.500
renders only, after re-cut	0.500
mixture, 29.5 % renders	0.760
camera label collapsed	0.769
camera and phase labels collapsed	0.500
mixture, class-first stratified draw (renders 30 % of draws)	0.681
scored on render rows / on real rows	0.500 / 0.791
domain indicator alone	0.578

After stratification the domain alone predicts engine state at 0.578. The remainder lies in the real half. Once windows are class-balanced, a real clip's flight phase predicts its engine state, because engines are off in orbit and on during ascent. The earlier "0.500 on real clips" came from a constant predictor on a draw that was 98 % lit, which the probe could not see. No sampler removes this and only more real clips that contain a change do. The training set has 104 such clips against 1,208 rendered ones, so about a fifth of all draws revisit those 104 clips. We state this as a limitation of the current data.

D Counterfactual pairs and rendering

The pair factory. A pair is two runs of the landing controller in the physics twin's landing environment. Both start from one seeded reset, with exact reset semantics, and differ in one commanded quantity, the altitude at which the landing burn ignites. The base altitude is drawn from $\mathcal{U}(900, 1300)$ m and rounded to 10 m; the intervention is drawn from $\{-300, -150, +150, +300\}$ m. We fly 200 pairs (400 episodes) at 60 Hz, deterministically from one seed, and choose the 40 held-out pairs before any is flown. The environment loads a twin record pinned by name. The pair check requires the members' altitudes to be identical before the earlier ignition, and they are on every pair. A usable pair has 2.5 s of descent before the earlier ignition (the conditioning frames) and 4 s after the later one (the consequence) resulting in 184 of 200 pairs to qualify. The interventions move the ignition by 1.4–3.0 s.

A withdrawn control. We had planned a second control, the choice of which engines relight. In the twin the controller flies its own ignition sequence, and overriding the choice left the trajectory bit-identical. A control that does not change the state cannot test command following, so we withdrew it before any video was generated.

Pairs anchored on the real flight. To connect pairs to real frames, we align the twin's clock to the flight through the displayed altitude. The real ignition time is read from broadcast frames (no flame at T+3,905.1 s, flame by 3,907.7 s). The factual member ignites at the altitude the twin has at that instant, and six of six seeds then ignite within 0.18 s of the frame-read time. Counterfactual members shift that altitude by ± 150 m.

Rendering. Unreal Engine 5.8.2[3] replays logged trajectories off screen at 60 Hz: Earth-fixed position and body attitude, per-engine thrust, gimbal and surface states. No physics runs inside it. The geodetic origin and every ground camera are anchored at the first replayed row. Anchoring at lift-off had placed a late replay thousands of kilometres from the origin, beyond double-precision world coverage, and it rendered an empty sky while the process exited normally. The pairs use two rigs, a chase camera and an on-board camera at the tip of the port forward flap (18 mm lens). An image-side test of broadcast against render decided port from starboard. Ground rigs mirror the broadcast's shot types (24, 35, 60 and 800 mm on a fixed 36×20.25 mm sensor). Rigs are *matched by shot type, mount point and focal length*, not solved as a camera calibration, and the on-screen display is not reproduced. Frames are 1280×704 at 30 fps.

Differences from the broadcast that remain. Exposure was the dominant artefact, and at first it masqueraded as physics. At a frame a reviewer described as "no flaps moving", the render had a mean luminance of 1.5 of 255, a silhouette. We measured per-rig exposure biases against a target 99th-percentile luminance where automatic histogram exposure was worse on 11 of 12 scenes and was rejected. The atmosphere above the sensible air is untuned, and a single rigid mesh cannot show stage separation. The booster's grid fins barely deflect in our physics, a modelling gap rather than a rendering one. For side-by-side review, each render is paired with a broadcast cut located by reading the display clock in the cut's frames. Rendered and real clips also differ in their *labels* unless care is taken (Appendix C.3, last paragraph), and in the base rates of Proposition 4.

D.1 The v5.0 factory and rigs

The release's 6 pair sets branch from the accepted physics twin of a named flight. Each holds 120 pairs after the visibility gate, 40 flown per intervention level. Two sets intervene on the ascent: an engine-out (one to six outer engines off at T+30–120 s) and throttle held at 0.4, 0.55 or 0.7 for 15 s. Two fix the boostback's engine set (11 to 3 engines) or its throttle (0.4 or 1.0). The last two hold the booster's landing ignition dark for 1, 2 or 3 s and fly the upper stage's landing burn on at most two of three engines. The pair check requires bit-identical altitude before the first effector difference, and that difference must fall within one control step of the plan. A set is kept only if the intervention is visible by eye on at least one rig in one pair. The far-side ascent engine-outs were invisible on every rig and were dropped. No existing rig resolved an engine-count change on a 33-engine stack, so we added an aft engine-bay rig on the skirt. The rigs are chase, hull-mounted, engine-bay and pad tower, at 1280×704 and 30 fps. A render process runs at about 20 frames per second in 2 GB of memory. 306 replay episodes of 12 s (up to four per flight, vehicle and phase) cover lift-off, ascent, hot staging with both vehicles, boostback, booster descent and landing, upper-stage ascent, coast, relight, entry and landing. The overlay drops out at staging on every flight. Hot staging and boostback are therefore bridged kinematics with the engine indicator measured, and each clip carries its bridged fraction. The renderer draws no entry plasma.

[3] `https://github.com/EpicGames/UnrealEngine/tree/5.8.2-release`

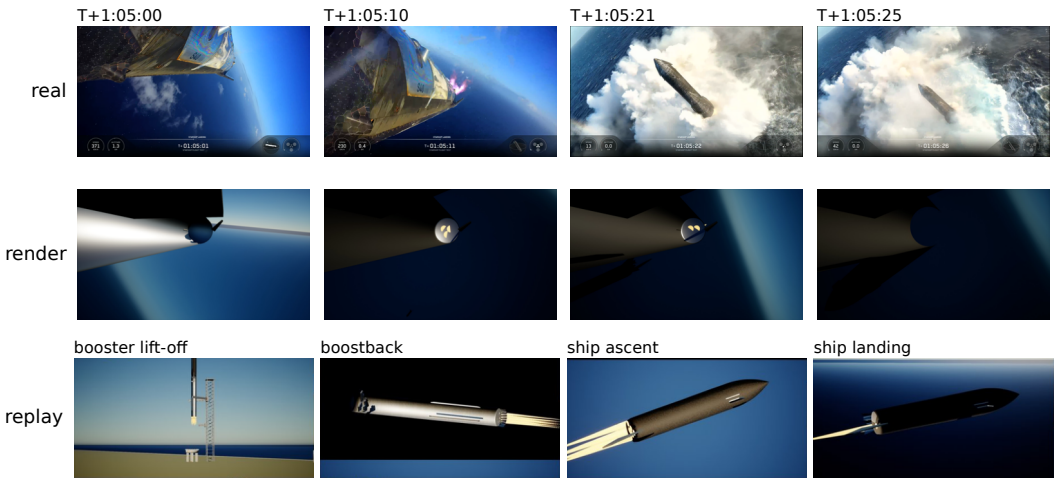


Figure 6: Broadcast frame (top) against the physics twin's render at the same mission time (bottom) on the tested flight's landing burn, and replay renders of four phases. Cameras are matched by shot type and focal length, not calibrated.

E World models

E.1 Cosmos 3 Edge: recipe

We post-train the released Cosmos 3 Edge checkpoint[4] in forward-dynamics mode and the 16B Cosmos 3 Nano[5] model shares the recipe on four accelerators. The launch vehicle is embodiment 31, the last free slot of 32. Only that slot's input and output projections are re-initialised, because loading with whole-tensor skips would reset the interface of every embodiment. The action row concatenates the value and validity of each live registry cell, 31 cells under the training registry (62 wide) and 32 under v5.0 (64 wide). Upper-stage commands come first. Cells are keyed by vehicle role and normalised by the same per-channel maps as the Predict2.5 embedder, and an absent cell is 0 with validity 0. The robot form keeps state cells valid on row 0 only, with commands on every row. A window is 61 frames at 10 fps (stride 3 on 30 fps footage), resized and centre-cropped to 832×480. Each variant's stratified sampling plan is recomputed at the 181-frame span.

The checkpoint holds 3.86 B parameters in total (safetensors). The recipe post-trains 1.4 B of the 3.4 B that form its mixture-of-transformers generator and Nemotron-2B language tower: the generator experts, the time embedder, the video-to-language and language-to-video projections and the action interface. The language tower, embeddings and vision encoder stay frozen. The loss is on vision only (`action_loss_weight` 0), so the inverse-dynamics head is untrained. Fused Adam runs at 10^{-4}, $\times 5$ on the action modules, with linear decay over the run. Training takes 6,000 iterations at 3.7–4.4 s per iteration on one accelerator, and the EMA weights are evaluated. The prompt is the JSON form the model was released with, with no idle frames and a fixed viewpoint token.

Pathway check. We compare the displacement of embodiment row 31, perpendicular to its initial value, with that of the 31 other rows, which must move by weight decay alone. On every variant the other rows match the closed-form decay prediction to five figures. Row 31 moves 0.014–0.037 by iteration 1,500, against 0.016 for an over-fit reference.

Cosmos 3 Nano. The 16B mixture-of-transformers model is post-trained in forward-dynamics mode with the launch vehicle as a new embodiment domain, whose input and output projections are initialised from scratch. As for Edge, only that domain's row of the shared projection tensor may be reset. In our run the new row moved and the other rows did not, a negative control inside one tensor.

[4] `https://huggingface.co/nvidia/Cosmos3-Edge`
[5] `https://huggingface.co/nvidia/Cosmos3-Nano`

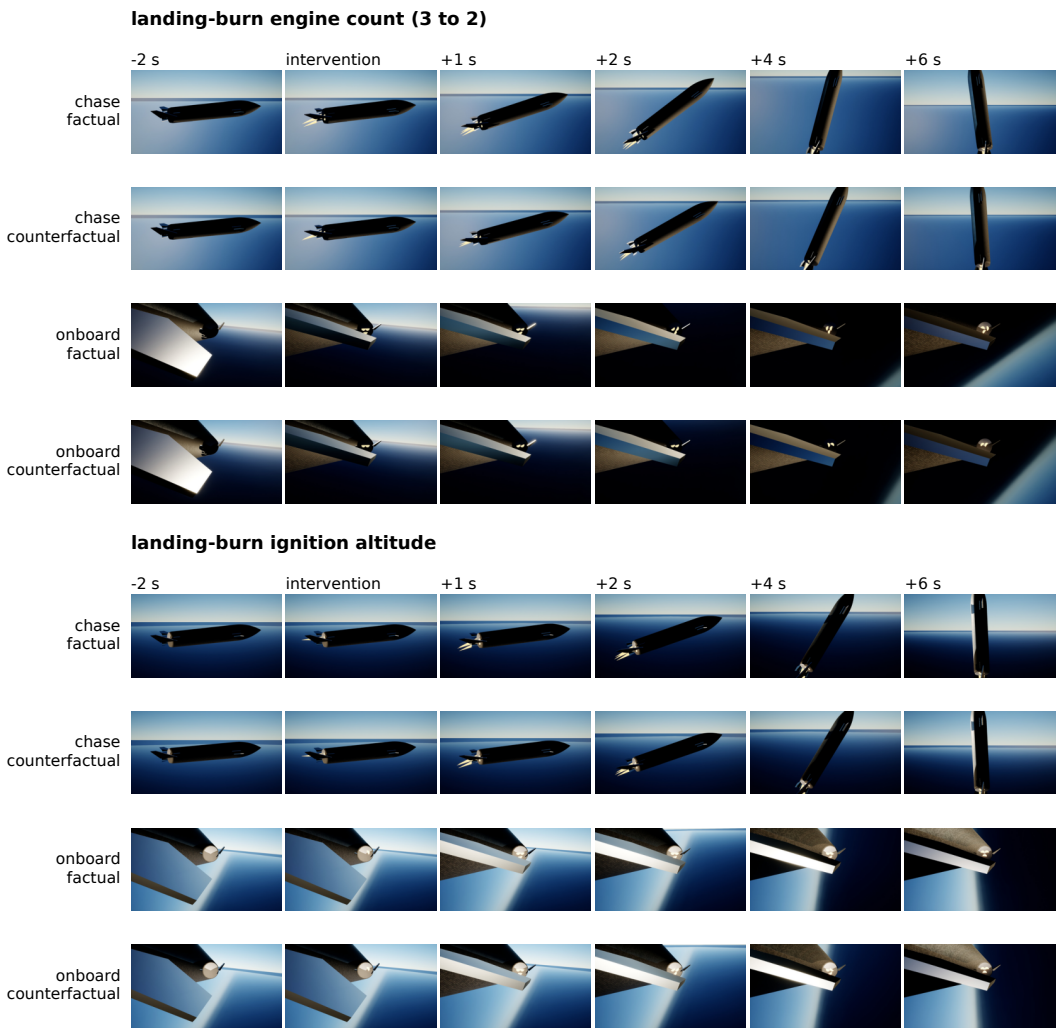


Figure 7: One held-out counterfactual pair (landing-burn ignition altitude, chase and hull-mounted rigs). Factual above, counterfactual below, six frames each. The trajectories are identical until the intervention.

Generation in this model is reproducible per rank and differs across ranks, so the conditions of a pair are generated on one rank.

E.2 COSMOS-PREDICT2.5-2B: RECIPE AND THE DIAGNOSED FIRST ATTEMPT

Predict2.5-2B. The backbone is the released 2B video diffusion transformer[6], with 32 blocks of width 2,048, latent compression $4 \times 8 \times 8$ and $1 \times 2 \times 2$ patches. It is trained with the rectified-flow objective

$$\boldsymbol{x}_\tau = (1-\tau)\boldsymbol{x} + \tau\boldsymbol{\epsilon}, \qquad \mathcal{L}(\theta) = \mathbb{E}\big\| f_\theta(\boldsymbol{x}_\tau, \tau, o_{1:P}, \boldsymbol{a}_{1:T}) - (\boldsymbol{\epsilon} - \boldsymbol{x}) \big\|^2, \tag{7}$$

where $\tau = 1$ is pure noise. We use its action-chunk-conditioned network through a thin subclass that overrides only the forward pass. Reading the released code ruled out two alternatives. An embedding inside the conditioner fails, because the conditioner overwrites a learned action embedding with the raw tensor on every step and must have no trainable parameters. Wrapping the network fails, because it renames every pre-trained parameter and silently breaks checkpoint loading. The network consumes $93 - 1 = 92$ action rows for 93 frames. Following the convention of its robot data, row t

[6] https://huggingface.co/nvidia/Cosmos-Predict2.5-2B

conditions the generation of frame $t+1$, and the conditioning frame receives no action. Our rows are per-frame readings, not differences, so we drop the last frame's reading and keep leading alignment. Clip cutting prepends frames before the target time; we remove the measured drift so that frame i and action row i share a time. The base model's clips run at 16 fps and ours at 30 fps, so a window of 93 frames spans 3.1 s.

Embedding. Per vehicle, the 24 normalised values and their mask, a vehicle-role code and a presence flag form a row of width 50, so two vehicles give a $92 \times 2 \times 50$ tensor. Each channel is normalised by a fixed symmetric-log or affine map whose constants are versioned in code, so no statistic is ever computed on data that could include held-out clips. A masked channel selects a learned embedding. The embedder is a two-layer perceptron whose output is added to the time embedding; its last layer is the pathway monitored below.

Recipe. Adapters of rank 32 (scale 32) train at learning rate 2^{-13} ($\times 10$ on the action modules) on 93 frames of 704×1280, for 6,000 iterations with 300 warm-up steps and a linear decay from 0.5 to 0.2 of the peak rate over the run. Each rank draws one clip per iteration. The M0 variant ran on one rank at batch 1 (about 31 accelerator-hours); the telemetry and H-full variants ran on two data-parallel ranks at batch 2 (H-full about 63 accelerator-hours). Each runs at about 19 s per iteration. The exponential moving average lags a freshly initialised layer, so pathway magnitude is read on raw weights.

One-variable variants. We cut the real-only training set out of the mixed one by hard links. It keeps the same videos, captions, labels, class and phase weights and scored channels; only the render entries are removed. On the windows the trainer actually draws, a validator re-derives the share of labelled cells per channel, the window class and change offset, and the context probe of Appendix C.4. The supplementary material prints the configuration difference of each pair of compared variants. The Edge variants of record differ only in the data path and the conditioning form. The Predict2.5 M0 variant also differs from H-full in rank count, and hence in batch (Recipe, above).

E.3 Training curves

F CommandBench

CommandBench is the paired protocol of Section 6.1, registered before any generation was scored.

F.1 Sets and generation

The pair factory of Appendix D flies the upper stage's landing burn from the accepted physics twin, on seeds disjoint from every training export. *ignite_time* shifts the ignition altitude by ± 120 m ($n = 64$; 33 positive and 31 negative shifts). *level_on* sets a late ignition against the factual one and is read by onset ($n = 61$); three pairs were dropped because the reference render's onset missed its flown event. *level_off* sets an early cut-off and is read by burn duty ($n = 4$). It lost 50 pairs because the cut-off render ends inside the window, and 10 that fell below the margin. Two smaller diagnostic sets from an earlier factory (± 150 m, $n = 19$; ± 300 m, $n = 15$) are reported beside them. Generation prepares frame 0 as training does and packs the set's own sidecars under the variant's registry. It uses a neutral caption at text guidance 0, 30 sampler steps, 61 frames and the exact sampler on a single rank. All conditions of a pair run in one process, because generation is deterministic within a process only.

Booster landing-burn sets (phase 2). Two further sets take the same 10 held-out ignition-delay pairs from the v5.0 booster landing factory (Appendix D.1). *booster_ignite_time_v1_c3*, the primary set, renders them on the engine-bay rig; *booster_ignite_time_v1_c3_chase* renders them on the chase rig. Every shift is a delay. The conditions are TRUE, SWAP, SELF and NULL. The oracle wins 10 of 10, and the blind and inverted stand-ins win none. The shuffled gate applies only where the commanded signs are mixed, so it is not applied here and the shuffled stand-in wins 10 of 10. The judge has been validated on blind labels of generations on either rig: every booster read so far used the render-only variant's validation on upper-stage generations. Its pass rule is at least 9 of 10 on the engine-bay rig, with SELF not passing.

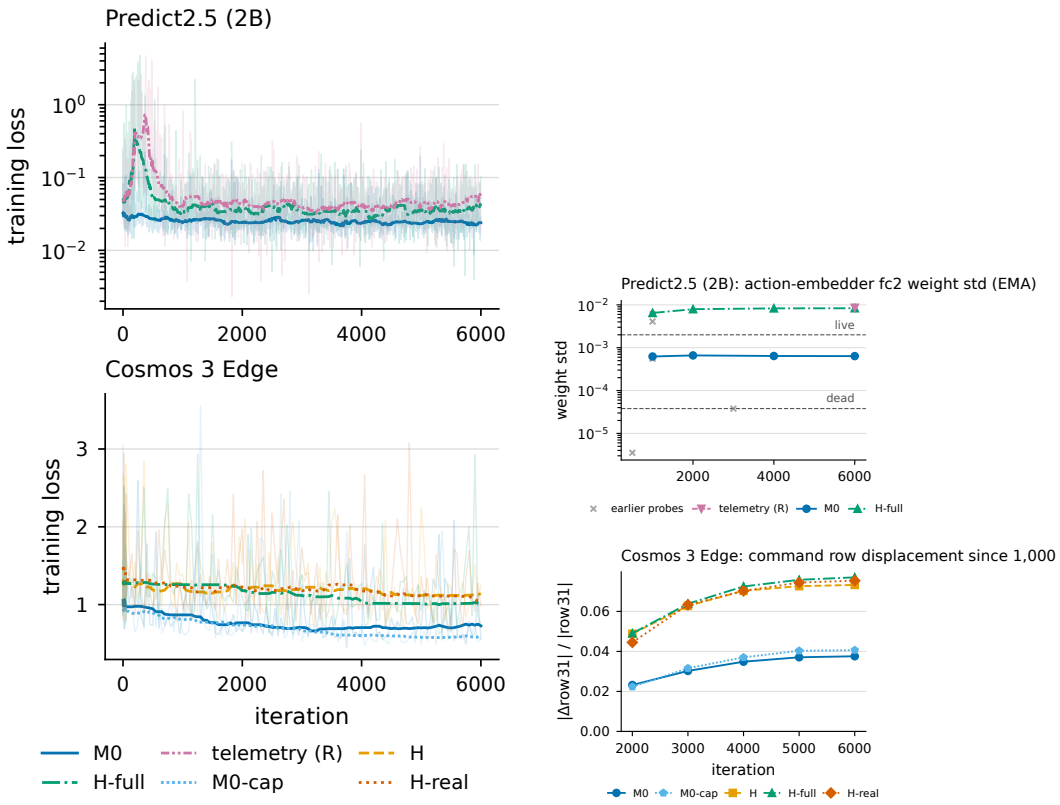


Figure 8: Training loss per iteration (left) for the variants of record (Edge M0, M0 with varied captions, H, H-full, H-real; Predict2.5 telemetry, M0, H-full). Pathway magnitude per checkpoint (right). For Predict2.5 it is the action embedder's output-layer standard deviation (EMA weights), with the dead and live references; for Cosmos 3 Edge it is the relative displacement of the launch vehicle's embodiment row since iteration 1,000. The variants that follow commands (M0, M0-cap) move least.

F.2 Judge validation

The warm-pixel judge thresholds the excess of warm-coloured pixels over the conditioning frame at 2×10^{-4} per frame. Onset is the start of the first run above threshold, and duty is the fraction of frames above it. We validated it on our own annotated blind labels of each model's own generations, at that model's geometry, against a bar of 0.90 for generations and 0.95 for renders. It scores 0.995 on the render-only variant, 1.000 on the mixed variant and 1.000 on renders. The full-window and real-only variants are read with the mixed variant's validation and flagged as such. A model without a passing validation scores nothing. A vision–language judge (Qwen3-VL-8B) scored 0.835 and is not used.

Each set passes its known answers before any model is scored. The oracle must reach at least 0.95 and the blind stand-in at most 0.05. The inverted stand-in must not win, and where the commanded signs are mixed the shuffled stand-in must not pass. A composite follower, the render plume difference pasted onto the model's own TRUE generations, must win at gain 1 and it wins 15/15 and 19/19 on the diagnostic sets. On renders the onset timing error is 0.033 s median and 0.10 s maximum, and the response error is at most 0.067 s. The 0.2 s margin is therefore three times the largest response error.

F.3 Statistic

Δ_c = read-out(intervened render) − read-out(reference render), Δ_g = read-out(SWAP) − read-out(TRUE), and gain = Δ_g/Δ_c. Wins are scored as in Section 6.1, with a margin of 0.25 on duty. A pair is a loss when the sign is opposite by at least the margin, and otherwise a tie. A variant *follows*

a set when the interval on its wins over all pairs lies above $\frac{1}{2}$ and SELF's does not. Every number is printed beside SELF's wins and losses, the NULL read-outs and, for the full-window variant, the STATE swap. Beside these stand the median gain with a bootstrap interval and a one-sided Wilcoxon p, the pairs with no read-out, the sign balance of Δ_c and the loss probe on the same checkpoint. Follow claims across variants are Holm-corrected. The headline variant is the one that follows more primary sets, with ties broken by median gain on *ignite_time* where both candidates are printed whatever the outcome.

F.4 THE LOSS PROBE

The loss probe, the secondary column, asks the model's own training loss whether the intervened video is cheaper under its own commands than under the reference commands, and vice versa. It runs at pinned noise levels $\sigma \in \{0.5, 0.8\}$, with a generator re-seeded per pair. The score is $(L[I \mid R] - L[I \mid I]) + (L[R \mid I] - L[R \mid R])$, so any caption main effect cancels and only the command–video interaction survives. A *text control* applies the same algebra to captions that name the engine state, and it must move on a model whose text channel is live. The untrained base moves on it (12/15) and not on commands. The probe is bit-identical across re-runs and takes about five minutes per checkpoint.

F.5 REAL FOOTAGE

The real-footage test is report-only. It takes the upper stage's landing burn from its aft hull camera, in pre-ignition windows with the engines off for 6 s, and commands the model to ignite at 1.0 s. The statistic is paired detection against TRUE and NULL, per flight. It is read only after the judge passes on blind labels of real-conditioned generations, and the judge has not passed. In the third labelling round, the we labelled 150 frames of the render-only and mixed variants' continuations of real windows. Only three plumes were drawn, all by the mixed variant; the judge read 0 of 3 and fired on 10 % of the mixed variant's no-plume frames. The row is therefore "not read". The 29 pre-ignition clips of an earlier protocol were unmeasurable, with the plume out of view in all 29.

F.6 EARLIER INSTRUMENTS

Paired statistics. For each comparison we report wins, losses and ties, the win rate with an exact Clopper–Pearson 95 % interval, and an exact sign test. We also report the mean paired difference with a paired t test, and the smallest effect detectable at 80 % power. That effect comes from the exact non-central t distribution, because the normal approximation is biased at our sample sizes. Unlike CommandBench, these statistics exclude ties from both the win rate and the sign test, so the point estimate and its interval share a denominator. SWAP donors are fixed in configuration before results are seen. A win on state channels alone would be reported as retrieval, not control, because the mission clock and the kinematic channels nearly identify a clip.

The diagnostic pilot (Section 2). Table 13 gives the result. The SELF control passes, so the comparison is admissible. At $n = 12$ the win rate is inconclusive by the rule registered beforehand, and the finding rests on effect sizes. A seed change moves pixels 7.6 times more than removing every action does. A second backbone of a different architecture, whose action projection measurably moved in training, gave the same null.

Table 13: Diagnostic pilot on the telemetry-conditioned model: 12 held-out clips, one seed, design fixed before generation.

comparison	pixel difference (/255)	PSNR against truth	win rate
true actions vs. none (NULL)	3.03	+0.003 dB	7/12
true actions vs. another clip's (SWAP)	3.44	+0.009 dB	8/12 [0.35, 0.90]
true actions, second seed (SELF)	23.10	–	7/12 (control)

The ignition read-out, in four forms. (1) An onset detector with a 2.5 s baseline cannot see a command at 1.0 s in a 3.1 s window. (2) Warm-pixel fraction against a 0.4 s baseline fires on the

clear real ignition and misses one that is partly out of view. It gives no false onset on 29 real clips without ignition. It is *blind*, however, on the telemetry-conditioned model, which paints fire from its first generated frames. The baseline is then already warm, and 16 generations that visibly burn show no onset. (3) Excess over the conditioning frame fixes that but *splits detections by seed, not by command*: one seed's fire flickers and another's rises steadily. (4) The final form is a five-frame moving average of the excess over the conditioning frame, above 0.01 for three frames. Re-validated, it fires at 2.03 s on the clear real ignition (visible at 1.57 s), misses the partial one, and gives no false onset on the 29 clips at thresholds 0.01–0.03. On chase-view renders a domain-specific threshold recovers twelve of twelve known ignitions within 0.07 s. No model was scored before the form was fixed. Thresholds are set per domain and validated on that domain.

A colour-free instrument. A video-to-video route rendered plumes grey, so we also use *divergence onset*. It is the first sustained departure of the per-frame mean absolute difference between the two videos of a pair from its level in the first 0.4 s, applied identically to generations and renders. On renders it fires on 24 of 24 pairs at 0.83–1.07 s for a command at 0.50 s. That lag is the detector's own hold-time bias, and comparing like with like subtracts it.

Instruments we rejected or bound. A learned regressor of altitude and speed from display-free frames reaches a held-out relative error of 20 %, which bounds any learned judge trained on this footage alone. Its within-clip change has the wrong sign as often as not, and we do not use it. Optical-flow motion correlates negatively with vehicle speed, because camera work dominates. A plume classifier scored below chance on generated video. The vehicle-axis angle from a segmentation mask correlates at 0.67 with the rendered attitude, below our bar of 0.9, and is report-only. On real frames no held-out clip of the landing contains the ignition itself (0 of 122). This is why the real-frame test commands an ignition into clips that have none, and why its read-out is report-only.

Fidelity metrics. PSNR, SSIM and LPIPS are computed against the real continuation under one resolution policy for all variants. The content-debiased Fréchet distance uses a fixed evaluation bench (47 clips, three seeds in the diagnostic runs) and prints the real-against-real value beside it. The display reader also applies to *generated* frames that include the display region. It reads generated digits at full confidence, and shows that one backbone freezes the display while the other advances its clock at the wrong rate. We use this only as a qualitative check.

G Results in full

G.1 Command following, all sets, seeds and backbones

Table 14 lists every scored checkpoint. Video scores exist for one training seed of each Edge variant. The loss probe (*p1_45*) scores a checkpoint without generation: it compares the denoising loss of the reference and the intervened render under the model's own commands, so a positive normalised effect means the commanded render is the likelier continuation and a text control that swaps captions instead of commands must stay flat.

G.2 Ablations

State visibility. The full-window variant follows more weakly than the robot-form variant (44/0/20 against 56/0/8) and its STATE swap wins 0/64; 1/61: window state does not substitute for the command. The telemetry variant of the same backbone (full-window state from displayed readings) is null on the probe while its caption control is live.

Renders. Removing real footage raises the effect (M0) while removing renders removes it (H-real, which draws no plume on 57 of 64 pairs).

Dose. Checkpoints the level_on probe effect of the render-only variant runs -0.0005, $+0.0041$, $+0.0086$, $+0.0102$, $+0.0157$, $+0.0197$ from 1,000 to 6,000 iterations, significant from 2,000, and of its captioned retrain -0.0006, $+0.0020$, $+0.0080$, $+0.0117$, $+0.0195$, $+0.0371$and the mixed, full-window and real-only variants stay within ± 0.003, the spread between two runs of the probe on one checkpoint, so their probe reads are not trends. The untrained base prefers the wrong command on ignite_time (2/19).

Table 14: CommandBench per checkpoint. Video sets: wins/losses/ties on ignite_time ($n = 64$), level_on ($n = 61$) and level_off ($n = 4$); SELF on the first two. Diagnostic v1 sets ($n = 19$ and 15): wins. Loss probe: wins and median normalised effect on level_on ($n = 15$) and ignite_time ($n = 19$). Booster sets: engine-bay rig; chase rig, $n = 10$ each, with SELF and the number of pairs on which NULL draws a plume. Cosmos 3 Edge at 6,000 iterations unless stated.

variant	video sets (ignite; level_on; level_off)	SELF	loss probe	note
Upper-stage landing burn				
M0, seed 42	64/0/0; 61/0/0; 2/4	5/11/48; 0/11/50	15/15, $+0.020$; 17/19, $+0.028$	v1 sets 19/19; 15/15; positive control
M0, seed 1	–	–	14/15, $+0.037$; 19/19, $+0.059$	probe only,
M0, seed 2	–	–	15/15, $+0.044$; 19/19, $+0.080$	probe only,
M0-cap	64/0/0; 61/0/0; 0/4	3/–/61; 0/–/61	15/15, $+0.037$; 19/19, $+0.047$ (text control 10/15)	provisional: judge read with M0's validation; pre-registration unapproved
H, seed 42	56/0/8; 56/0/5; 0/4	18/26/20; 8/27/26	inconclusive	v1 sets 17/19; 14/15; headline
H, seed 1	–	–	13/15, $+0.005$; 14/19, $+0.005$	probe only,
H-full	44/0/20; 53/0/8; 1/4	18/25/21; 7/26/28	inconclusive	v1 sets 14/19; 15/15; STATE swap 0/64; 1/61
H-real	0/2/62; 0/0/61; 0/4	no read-out on 57/64; 57/61	inconclusive; text control 11/15	v1 sets 0/19; 0/15; no plume on 57/64
telemetry (M1)	–	–	6/15, -0.0002 (text control 12/15)	command null, caption control live;
untrained base	–	–	5/15, -0.001; 2/19, -0.009 (text control 12/15)	instrument check
Nano M0 @1,500	–	–	13/15, $+0.022$; 18/19, $+0.028$	16B backbone, replicates Edge M0
Nano H @1,500	–	–	12/15, $+0.003$; 12/19, $+0.001$	level_on follows ($p = 0.005$)
Booster landing burn, ignition delay				
M0-v5, seed 1	10/0/0; 10/0/0	2/0/8; 1/2/7	engines 8/10, chase 10/10 ($+0.050$)	NULL plume on 2; 0 pairs
M0-v5, seed 2	8/0/2; 6/0/4	2/3/5; 1/2/7	engines 7/10, chase 8/10	2 pairs without read-out; NULL plume on 6; 6
H-v5 (mixed)	0/0/10; 0/0/10	2/5/3; 0/0/10	booster 4/10	4; 9 pairs without read-out; where read, TRUE, SWAP and NULL agree
Cosmos-Predict2.5-2B				
telemetry (R)	–	–	8/15 [0.27, 0.79]	text control wrong sign; pathway 8.4×10^{-3}
M0	–	–	9/15 [0.32, 0.84]	pathway 6.3×10^{-4}, below the 10^{-3} bar
H, M1	–	–	–	never evaluated: checkpoints on
H-full	–	–	–	not scored: no time-aligned harness route

Seeds. The render-only dose trajectory replicates on two further training seeds (14/15, $+0.037$; 19/19, $+0.059$; 15/15, $+0.044$; 19/19, $+0.080$) and the mixed variant's second seed does not replicate on EMA weights at 6,000 and was not video-scored.

Scale. The 16B backbone at a token-matched 1,500 iterations replicates the render-only effect to two figures.

Weights. EMA and raw weights agree on every Edge variant and they disagree only on the 16B mixed variant.

Captions. The render-only variant's captions were two template strings, so its flat text control is not by itself evidence that the command effect is specific. A retrain on the same clips with 938 distinct scene-only captions (no propulsion words) follows on every pair (64/0/0, 61/0/0; SELF 3/–/61; 0/–/61), with a probe effect 1.7–1.9$\times$ the template variant's and cd-FVD 1,234 $\pm$ 59 against 1,291 $\pm$ 48. That says that the command effect does not depend on caption starvation. Its bench caption is the template variant's training string, so the two variants are not a clean one-variable pair, and its judge read borrows the template variant's validation.

Pathway. Embodiment-row displacement at 6,000 iterations: M0 3.8 %, M0-cap 4.1 %, H 7.3 %, H-full 7.7 %, H-real 7.5 %while the other rows move 2.1 % on every read, entirely by weight decay. Displacement is necessary for following and does not track it.

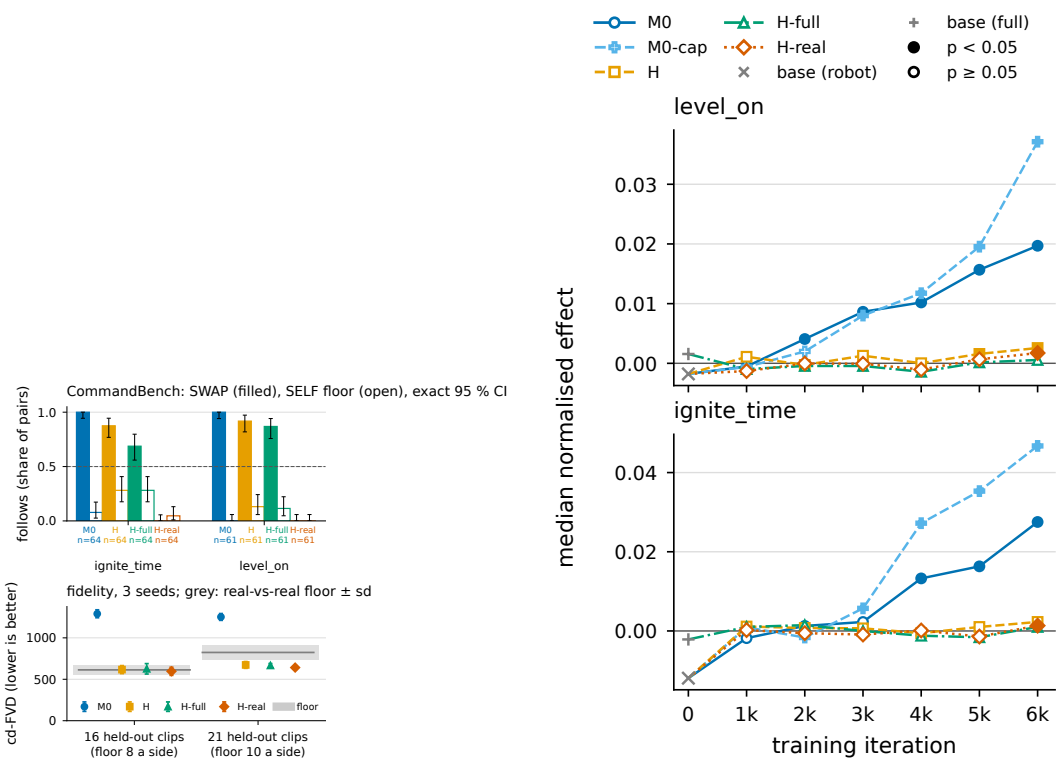


Figure 9: Left: follow fraction with exact intervals on *ignite_time* and *level_on* for the four Edge variants, SELF beside each, and cd-FVD against the real-versus-real floor. Right: loss-probe effect against training iteration for the variants of record (checkpoints at every 1,000 iterations). Filled markers are significant at $p < 0.05$ (Wilcoxon).

Real frames (descriptive). On two held-out real windows of the upper stage just before its landing burn, with the ignition commanded at 1.0 s, the mixed variant develops a glow from about 4 s and the full-window variant an exhaust cloud, the real-only variant does not change, and the render-only variant leaves the footage at once (Figure 10). On three held-out real windows of the booster's landing burn, one containing the actual ignition, no booster variant reproduces the flame under the recorded command, a command moved 2 s later, or no command (Figure 11).

G.3 FIDELITY, SECOND BENCH

Table 15: Fidelity on the 21-clip bench (three seeds); real-versus-real floor 824 ± 90.

variant	PSNR	SSIM	LPIPS	cd-FVD
M0 (renders)	13.66	0.501	0.578	1,252 ± 41
M0-cap	13.88	0.491	0.564	1,184 ± 40
H (real + renders)	17.58	0.676	0.339	673 ± 41
H-full	17.83	0.681	0.336	667 ± 23
H-real	17.90	0.684	0.331	641 ± 24

No fidelity was measured for the booster variants.

G.4 THE VIDEO-TO-VIDEO ROUTE

Driving a released video-to-video model with the twin's counterfactual render as its structural control follows the render's geometry (edge recall 0.89 against 0.61 for the other member, on 108 of 108 pairs) and the intervention's timing to 0.00 ± 0.00 s on the hull-mounted rig, yet moves the clips

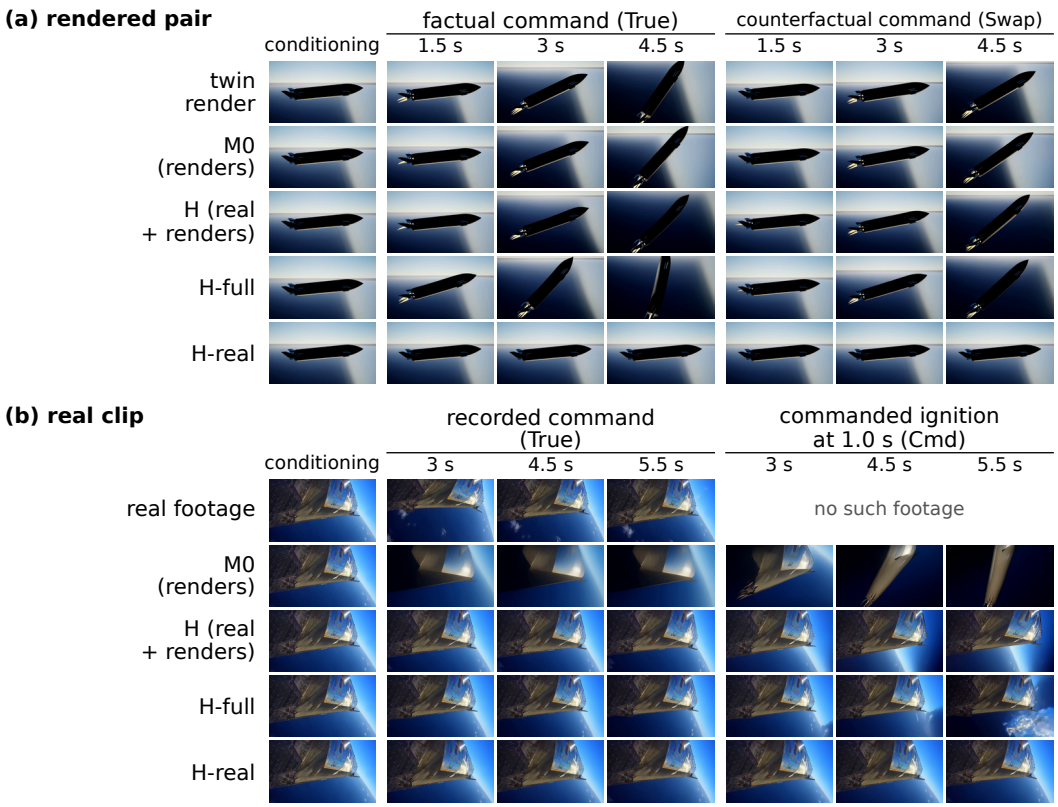


Figure 10: Generations of every variant of record. (a) The held-out rendered pair of Figure 2 with the full-window variant added: the ignition moves between the TRUE and SWAP columns in the render-only, mixed and full-window variants and not in the real-only variant. (b) A held-out real window of the upper stage just before its landing burn (the recorded ignition falls after the window): the real footage, and each variant's generation from the same conditioning frame under the recorded command (TRUE) and with the ignition commanded at 1.0 s (CMD), at 3, 4.5 and 5.5 s. Under CMD the mixed variant develops a glow from about 4 s and the full-window variant an exhaust cloud. The real-only variant does not change and the render-only variant leaves the footage at once under both commands. Descriptive only: the judge is not validated on real-conditioned generations.

away from real footage on cd-FVD (1,231 against 997 for the raw render and 589 real-versus-real, $n = 23$), and does not carry the plume's colour. It is an interesting result which was not further investigated and expanded upon.

G.5 PREDICT2.5 DIAGNOSTIC

Table 16: The diagnostic pilot on the telemetry-conditioned Predict2.5 model: 12 held-out clips, one seed, design fixed before generation.

comparison	pixel difference (/255)	PSNR against truth	win rate
true actions vs. none (NULL)	3.03	+0.003 dB	7/12
true actions vs. another clip's (SWAP)	3.44	+0.009 dB	8/12 [0.35, 0.90]
true actions, second seed (SELF)	23.10	–	7/12

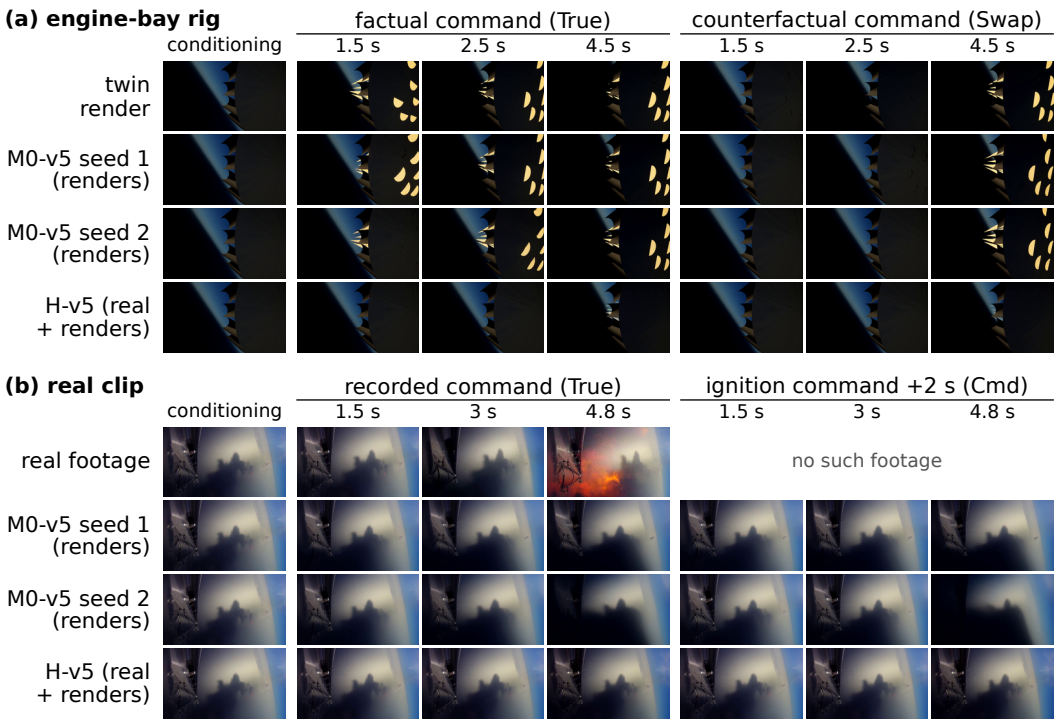


Figure 11: Booster landing burn, provisional read (Table 14). (a) One held-out rendered pair whose ignition is delayed by 2.1 s, engine-bay rig: the twin's renders (top) and each booster variant's generation from the shared conditioning frame under the factual (TRUE) and counterfactual (SWAP) command at 1.5, 2.5 and 4.5 s. Both render-only variants light the bay in the twin's order, factual first, counterfactual only by 4.5 s; the mixed variant lights it under neither. (b) A held-out real window of the booster's landing burn from its hull camera, with the real ignition at 4.5 s: the footage, and each variant's generation under the recorded command (TRUE) and with the ignition command moved 2 s later (CMD), at 1.5, 3 and 4.8 s. No variant reproduces the flame; one render-only seed darkens the scene instead. Descriptive only.

H COMPUTE

A cluster with two nodes was used (Table 17). The Edge variants were trained one accelerator each, in parallel on the four cards. The Nano variants were trained on four accelerators in 2 nodes in parallel. The Predict2.5 were trained in a single node using 2 accelerators per run.

Table 18 gives the measured cost of each workload.

Table 17: Hardware and software.

Compute Cluster (x2)	4× NVIDIA H200 NVL (143 GB), 192 CPU cores, 1.5 TB RAM, CUDA 13.0
physics, rendering pipeline	JSBSim 1.3.1; Unreal Engine 5.8.2 built from source, headless Vulkan Python 3.12, PyTorch 2.12; PaddleOCR, Grounding DINO, SAM 2.1, Cosmos-Embed1, PySceneDetect, Weights & Biases in offline mode

Table 18: Measured cost per workload. Accelerator-hours (acc·h) are wall time times cards.

Workload	Unit	Measured
JSBSim forward step	per core	1.2×10^5 steps/s
twin forward re-flight	per flight	0.4 s
twin ascent fit (search baseline, 3 seeds)	per flight	4 min on 12 cores
render, one 12 s episode	per rig, per editor	16–19 s (≈20 fps); 8 editors on 4 cards
render, released set (1,244 clips)	total	≈7 editor-hours
Cosmos 3 Edge variant, 6,000 it	one card	6.4–7.2 h (3.7–4.4 s/it)
Predict2.5 variant, 6,000 it, one rank	one card	31 acc·h (18.6 s/it)
Predict2.5 variant, 6,000 it, two ranks	two cards	63 acc·h (18.9 s/it)
Cosmos 3 Nano diagnostic, 3,000 it	four cards	54 acc·h
CommandBench, one Edge checkpoint	four cards	16–20 min generation (715 videos), 14 s scoring
fidelity (16 and 21 clips, 3 seeds), four variants	one card	14 min
loss probe, one checkpoint	one card	5–10 min
Predict2.5 pair bench rehearsal	per variant and seed	12 acc·h